IN-37026

p 236

DOE/NASA/0247-3
NASA CR-174983
PPI-FID-300086

WEST-3 Wind Turbine Simulator Development

Volume 3: Software

J.A. Hoffman and S. Sridhar
Paragon Pacific Inc.

July 1985

Prepared for
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Lewis Research Center
Under Contract DEN 3-247

for
U.S. DEPARTMENT OF ENERGY
Conservation and Renewable Energy
Wind/Ocean Technology Division

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Printed in the United States of America

Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

NTIS price codes[1]
Printed copy: A11
Microfiche copy: A01

[1] Codes are used for pricing all publications. The code is determined by the number of pages in the publication. Information pertaining to the pricing codes can be found in the current issues of the following publications, which are generally available in most libraries: *Energy Research Abstracts (ERA)*; *Government Reports Announcements* and Index (GRA and I); *Scientific and Technical Abstract Reports (STAR)*; and publication NTIS-PR-360 available from NTIS at the above address.

DOE/NASA/0247-3
NASA CR-174983
PPI-FID-300086

WEST-3 Wind Turbine Simulator Development

Volume 3: Software

J.A. Hoffman and S. Sridhar
Paragon Pacific Inc.
Torrance, California 90503

July 1985

Prepared for
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135
Under Contract DEN 3-247

for
U.S. DEPARTMENT OF ENERGY
Conservation and Renewable Energy
Wind/Ocean Technology Division
Washington, D.C. 20545
Under Interagency Agreement DE-AI01-76ET20320

CONTENTS

FOREWORD

The work documented in this report was sponsored by the NASA Lewis Research Center, under Contract DEN3-247. The work was performed by Paragon Pacific Inc., 530 Maple Ave., Torrance, CA 90503. The NASA Project Manager was Mr. D. C. Janetzke.

Principal participants in the contract activity were J. A. Hoffman, D. Weisz and S. Sridhar of Paragon Pacific Inc., and Dr. M. D. Campbell of Talisman Systems, Drawer CP255, Manhattan Beach, CA 90266.

SUMMARY

This report presents ,a description of the software developed for WEST-3, a new, all digital, and fully programmable wind turbine simulator developed by Paragon Pacific Inc.. In addition to the usual complement of software found in any digital computer, WEST-3 has additional software modules specialized for the processing of wind turbine simulation models. In the report, the process of wind turbine simulation on WEST-3 is described in detail.

The major steps are, the processing of the mathematical models, the preparation of the constant data, and the use of system software to generate executable code for running on WEST-3. The mechanics of reformulation, normalization, and scaling of the mathematical models is discussed in detail; in particular, the significance of reformulation which leads to accurate simulations. Descriptions of the preprocessor computer programs which are used to prepare the constant data needed in the simulation are given. These programs, in addition to scaling and normalizing all the constants, relieve the user from having to generate a large number of constants used in the simulation. Also given in the report are brief descriptions of the components of the WEST-3 system software: Translator, Assembler, Linker, and Loader. In common with other digital computers, the task of the WEST-3 system software is to generate the code which can be loaded into the machine.

The report contains details of the aeroelastic rotor analysis, which is the center piece of a wind turbine simulation model. Listings of the variables, constants, and equations used in the simulation are also given.

1. INTRODUCTION

Paragon Pacific Inc. has developed the WEST-3 for the real-time domain simulation of wind turbines. WEST-3 is an all digital, fully programmable, parallel processing system. It avoids the "hardwired" analog implementation of the previous WEST systems. High speed simulation is made possible by the parallel processing capability, and full programmability is assured by the all digital technology. A complete description of the WEST-3 system can be found in References 1 and 2. The details of the WEST-3 software are presented in this report.

The overall performance and usefulness of any simulation system depends upon the effectiveness of the software; and WEST-3 is no exception. The primary objective of the software developed for WEST-3 is to make it easier for a user to conduct useful wind turbine simulations. To meet this objective, in addition to the usual complement of software found in any digital computer, WEST-3 has additional software modules specialized for the processing of wind turbine simulation models.

Figure 1.1 shows an overall schematic of the WEST-3 simulation process consisting of,

1. preparation of the simulation model (rotor and subsystems)
2. preparation of the constant data
3. use of the system software to generate executable code
4. running of the simulation and the inevitable debugging.

Presented in this report are descriptions of the various components of software required to carry out a wind turbine simulation on WEST-3. The report closely follows the schematic of Figure 1.1, and is organized as follows:

Section 2 contains descriptions of all the mathematical models used in the simulation. Reformulation, normalization and scaling, which are of critical importance in WEST-3 simulations, are discussed in Section 3. A discussion of variable scale factors is presented in Section 4. Subsequent sections deal with the preparation of constant data, and the system software available in WEST-3. Section 7 contains some concluding remarks. A list of References is given in the last section. Appendices contain some important details of the aeroelastic rotor analysis, gimbal analysis, and complete listings of the variables, constants, and equations used in the simulation.

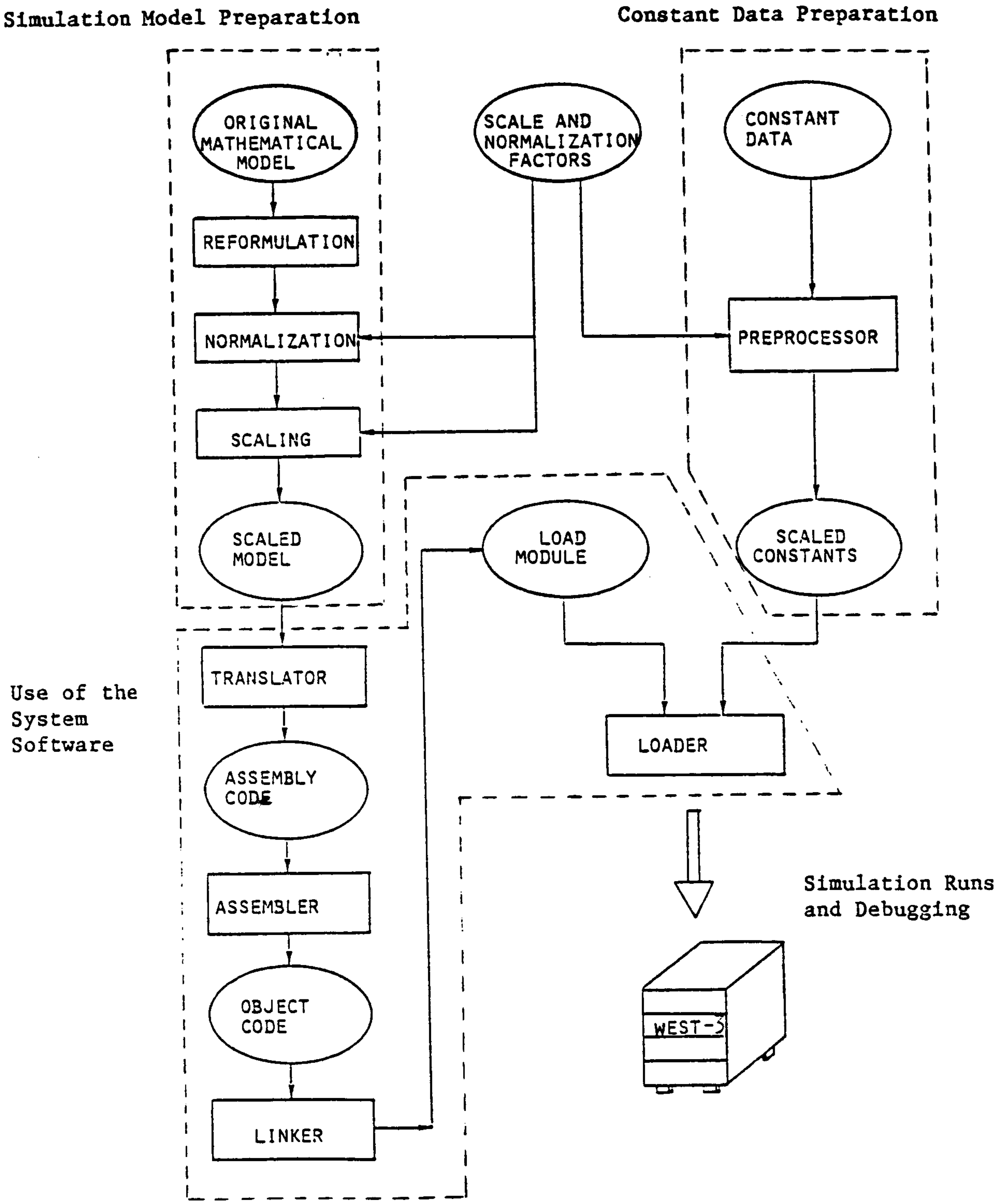

Simulation Model Preparation
Constant Data Preparation
ORIGINAL MATHEMATICAL MODEL
SCALE AND NORMALIZATION FACTORS
CONSTANT DATA
REFORMULATION
NORMALIZATION
PREPROCESSOR
SCALING
SCALED MODEL
LOAD MODULE
SCALED CONSTANTS
Use of the System Software
TRANSLATOR
LOADER
ASSEMBLY CODE
ASSEMBLER
Simulation Runs and Debugging
OBJECT CODE
WEST-3
LINKER

Figure 1.1. Overall Schematic of the WEST-3 Simulation Process

2. WIND TURBINE MATHEMATICAL MODEL

The simulation model of a wind turbine is a mathematical description of a number of complex physical phenomena involved in the extraction of useful energy from the system. The full model is formed by combining a number of smaller models of the various physical systems in the problem. Figure 2.1 shows the overall configuration of the simulation model. It is a synthesis of models for the rotor, the air mass, the support (tower), the power train, and the control system. Figure 2.2 is a more detailed schematic of the simulation model showing the variables that are passed among the subsystems.

Appendix A defines the general nomenclature conventions used in this report. Some of the nomenclature used in this report differs from conventional practice. Therefore, a study of Appendix A is strongly recommended for an understanding of the equations presented in this report. Also Table B.1 of Appendix B contains the nomenclature pertinent to the rotor model. A definition of all the variables and constants used in the simulation model can be found in Appendix F.

Models presented herein have been used in other wind energy system simulations (see, for example, References 3, 4 and 5). The descriptions of the models presented in this report are complete, and should not require references other than standard texts on mechanics and controls. However, References 3, 4 and 5 do present useful data on more detailed versions of these models. Some of the major features of the model implemented in WEST-3 are,

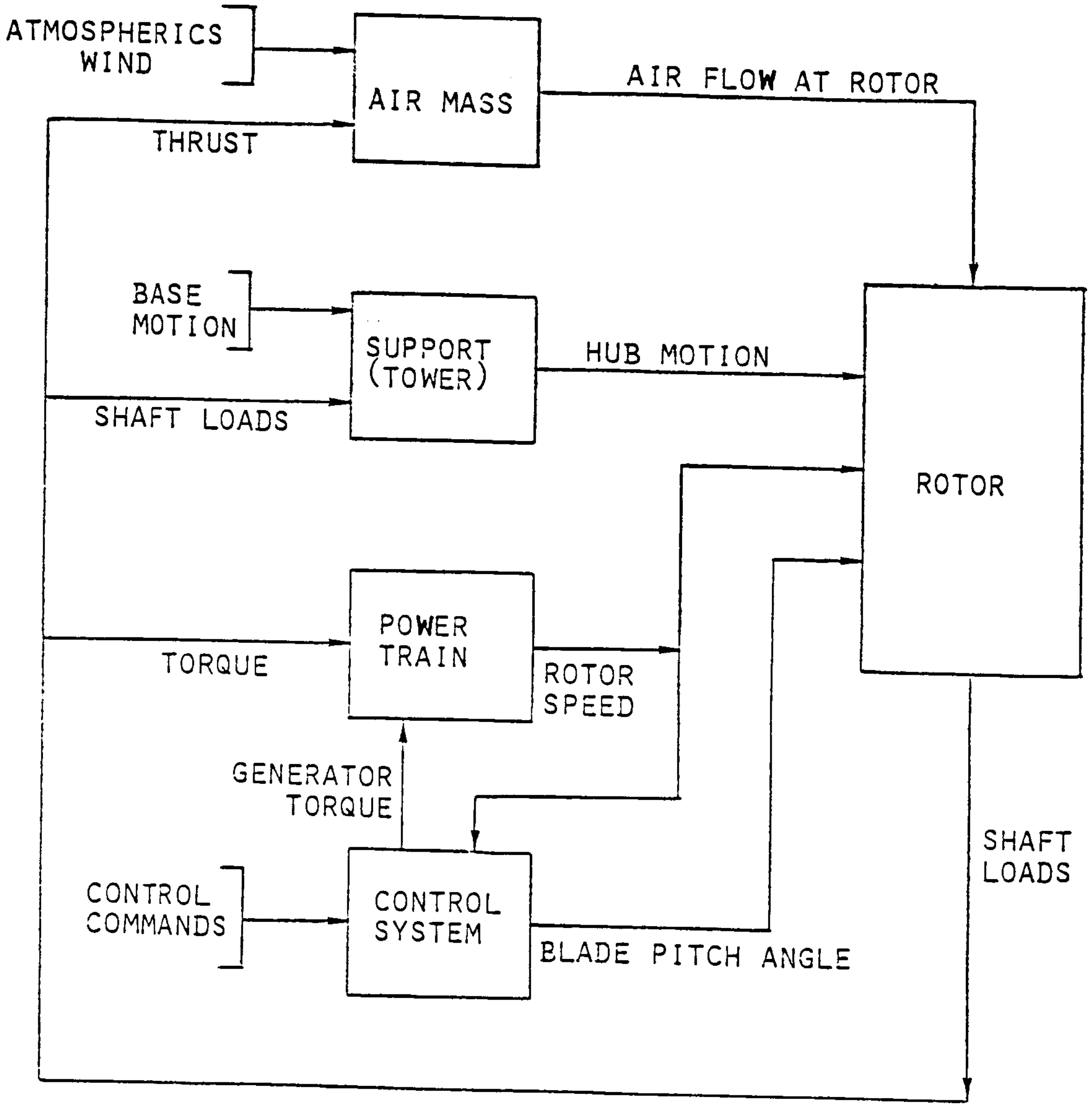
1. Three elastic degrees of freedom for each blade
2. Tower model with six physical (three modal) degrees of freedom
3. Gimballed/Teetering rotor capability
4. Blade tip loss, and flow retardation effects in the rotor
5. Nonlinear wind shear model
6. Bandpass wind gust filters
7. Tower shadow model
8. General models for the power train and control system

Due to the modular nature of the implementation, and the full programmability of WEST-3, alternate models/tasks can be incorporated with relative ease. Following are descriptions of the subsystems shown in Figures 2.1 and 2.2.

2.1 Aeroelastic Rotor

Of all the subsystems, the rotor is by far the most complex accounting for a bulk of the computational load in the simulation. A detailed description of the aeroelastic rotor model is given in Appendix B. Figure 2.3 shows a simplified schematic of the rotor model. Following is an outline of the major tasks performed in the rotor model, during numerical integration, for every azimuthal position of the rotor.

* Compute the aerodynamic loads on the blade by using strip theory. Each blade is divided into a number of radial segments and radial stations. The aerodynamic coefficients, as functions of the angle of attack, are obtained by table look-up.



Fᶦgure 2.1. Simplified Schematic of the Wind Turbine Simulation Model

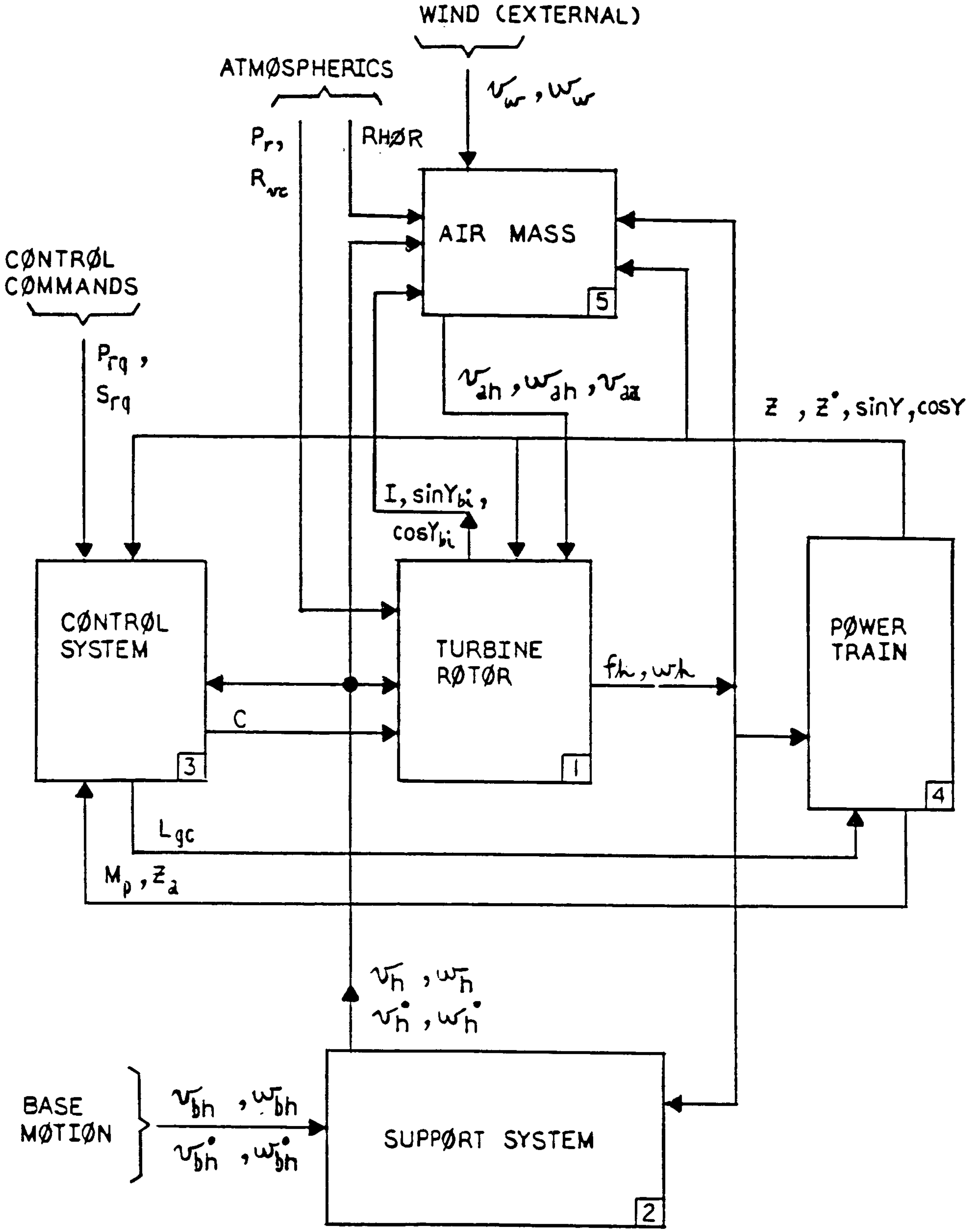


Figure 2.2. Flow of Variables in the Wind Turbine Simulation Model

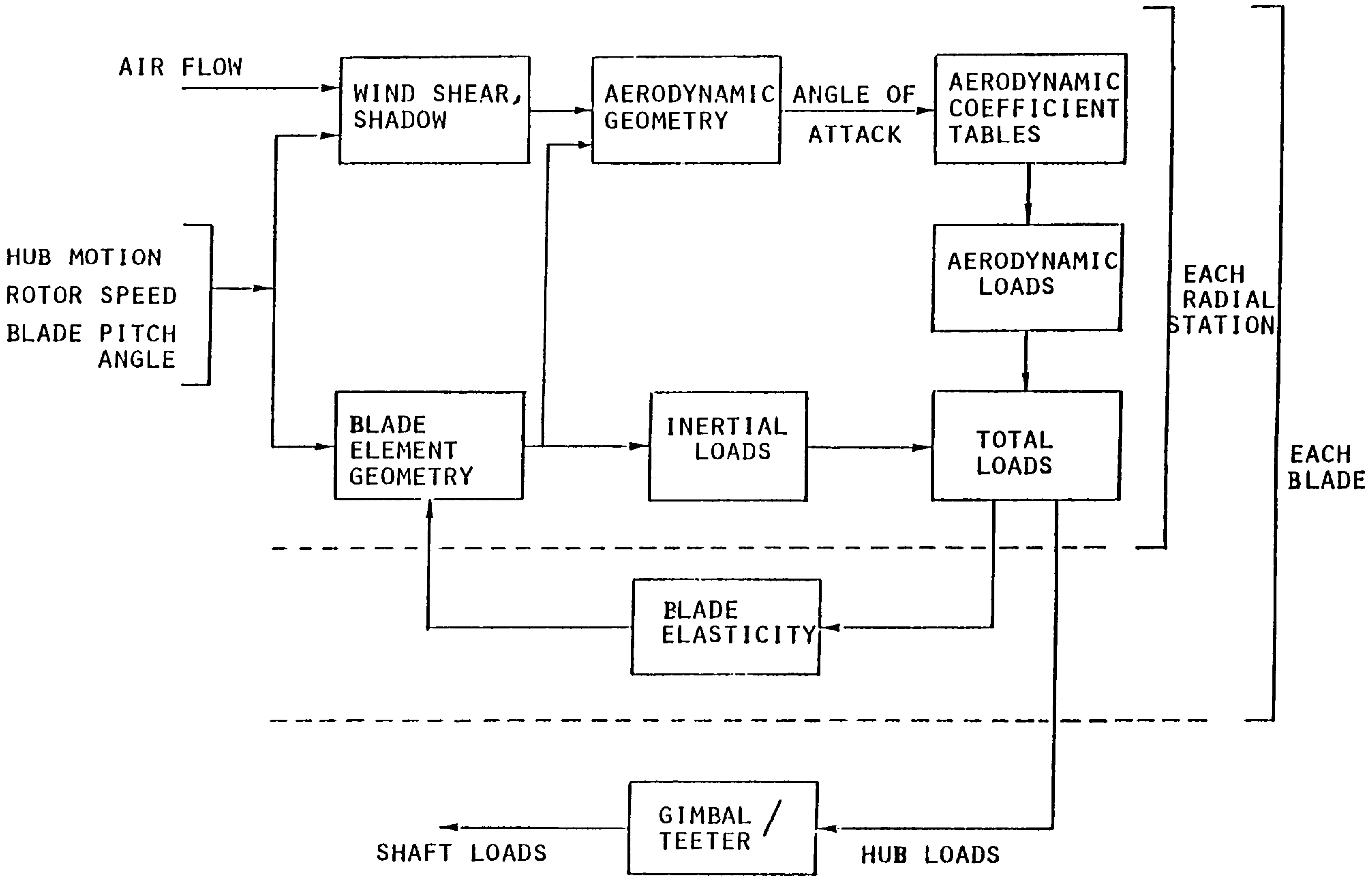


Figure 2.3. Simplified Schematic of the Aeroelastic Rotor Model

- Determine the effects of elasticity on the blade deflection, velocity, acceleration, and loading at each radial station.

- Obtain the forces and moments at the rotor hub by a summation of the loads at all the radial stations of the blade.

- Account for the effects of the rotor hub degrees of freedom, i.e., gimbal (teetering), if present in the model.

2.2 Support (Tower):

The following tasks are performed in the tower model:

- Simulate the dynamics of the tower/nacelle by using the modal analysis, the modal information being typically generated by a finite element program such as NASTRAN.

- Compute the velocity and acceleration of the hub for use by the rotor model.

It is not practical to hypothesize a fully general model for a subsystem such as the support system, because of the many possible designs. Accordingly, the approach taken herein is to implement a general linear model. Other nonlinear phenomena, such as hysteresis in gear trains, can be included in the model as and when required. The support system equations which can incorporate many configurations of tower/nacelle dynamics are,

$$s^{\cdot\cdot} + D_s \cdot s^{\cdot} + W_s^2 \cdot s = g_s,$$

where

$$g_s = G_{sf} \cdot f_h + G_{sm} \cdot m_h$$

The hub velocity and acceleration are given by,

$$v_h = S_v \cdot s^{\cdot} + v_{bh} \quad ; \qquad w_h = S_w \cdot s^{\cdot} + w_{bh}$$

$$v_h^{\cdot} = S_v \cdot s^{\cdot\cdot} + v_{bh}^{\cdot} \quad ; \qquad w_h^{\cdot} = S_w \cdot s^{\cdot\cdot} + w_{bh}^{\cdot}$$

A modal model has been used (see discussion of modal analysis in Appendix B). The vector, s, contains the normal coordinates. The vector, g_s contains the generalized forcing functions, each divided by its respective generalized mass. The diagonal operators, $D_s \cdot$, and $W_s^2 \cdot$ are damping and frequency matrices, respectively, associated with the normal mode analysis. The forces and moments acting on the tower are, respectively, f_h and m_h. The operators, $S_v \cdot$ and $S_w \cdot$ are the modal matrices, so that they produce the shaft motion when they operate on the normal coordinate velocity, $s^{\cdot}$.

Arrays $G_{sf} \cdot$ and $G_{sm} \cdot$ operate on the shaft loads produced by the rotor to generate the generalized forcing function. These operators can be derived by transposing the modal matrix operators followed by dividing each resulting row with the generalized mass of the associated mode.

Columns v_{bh} and w_{bh} are incorporated in the support system model, which are supplied to the model from an external source. These columns represent "base" motions of a moving system to which the elastic support system is attached (e.g., if a wind turbine is mounted on a barge). The external source can be the System IO Data Interface (SIDI) incorporated in the WEST-3 hardware, or it can be an additional set of models coded and loaded in WEST-3 to represent such a base flexibility. These additional columns have been incorporated in the model at this juncture to provide more generality.

A specialized version of the general matrix support models as presently implemented consists of a tower with three modal degrees of freedom, in the following directions: surge (motion along the #3 rotor axis), sway (lateral motion along the #2 rotor axis), and yaw (rotation about the #1 rotor axis, or tower torsion).

2.3 Control System:

The function of the control system is to change the blade pitch angle so that a desired rotor speed and/or power output is maintained. As is the case of the support system, control systems can take on many configurations. A general linear state controller model is implemented in WEST-3. This model can embrace many control system configurations, but will require additional code if nonlinear phenomena are to be included. The generic equations are,

$$a^{\cdot} + A_a . a = a_p M_p + a_z Z + a_{zd} Z^{\cdot} + a_g Z_a + a_{pr} P_{rq} + a_{sr} S_{rq} + a_o^{\cdot}$$

$$c = C_a . a + c_p M_p + c_z Z + c_{zd} Z^{\cdot} + c_g Z_a + c_{pr} P_{rq} + c_{sr} S_{rq} + c_o$$

$$L_{gc} = L_a^T . a + L_p M_p + L_z Z + L_{zd} Z^{\cdot} + L_g Z_a + L_{pr} P_{rq} + L_{sr} S_{rq} + L_o$$

The column, a, is the state vector for the system, and can be any size. Control system degrees of freedom associated with filters, compensation networks, integrators, etc., can be converted into the state form shown. The control system receives inputs, processes them with the state algorithms, and produces outputs. Inputs are: command requested power (P_{rq}), command rotor speed (S_{rq}), rotor speed (Z), rotor acceleration ($Z^{\cdot}$), power train moment (M_p), and alternator speed (Z_a), and the shaft velocities and time derivatives of velocities. The zero subscript denotes initial conditions or some constant offsets, if any. The control system produces the control column, c, for the rotor, and a torque on the alternator rotor, L_{gc}.

Figure 2.4 shows an example of a control system which is typical for a wind turbine; blade and alternator controls are included.

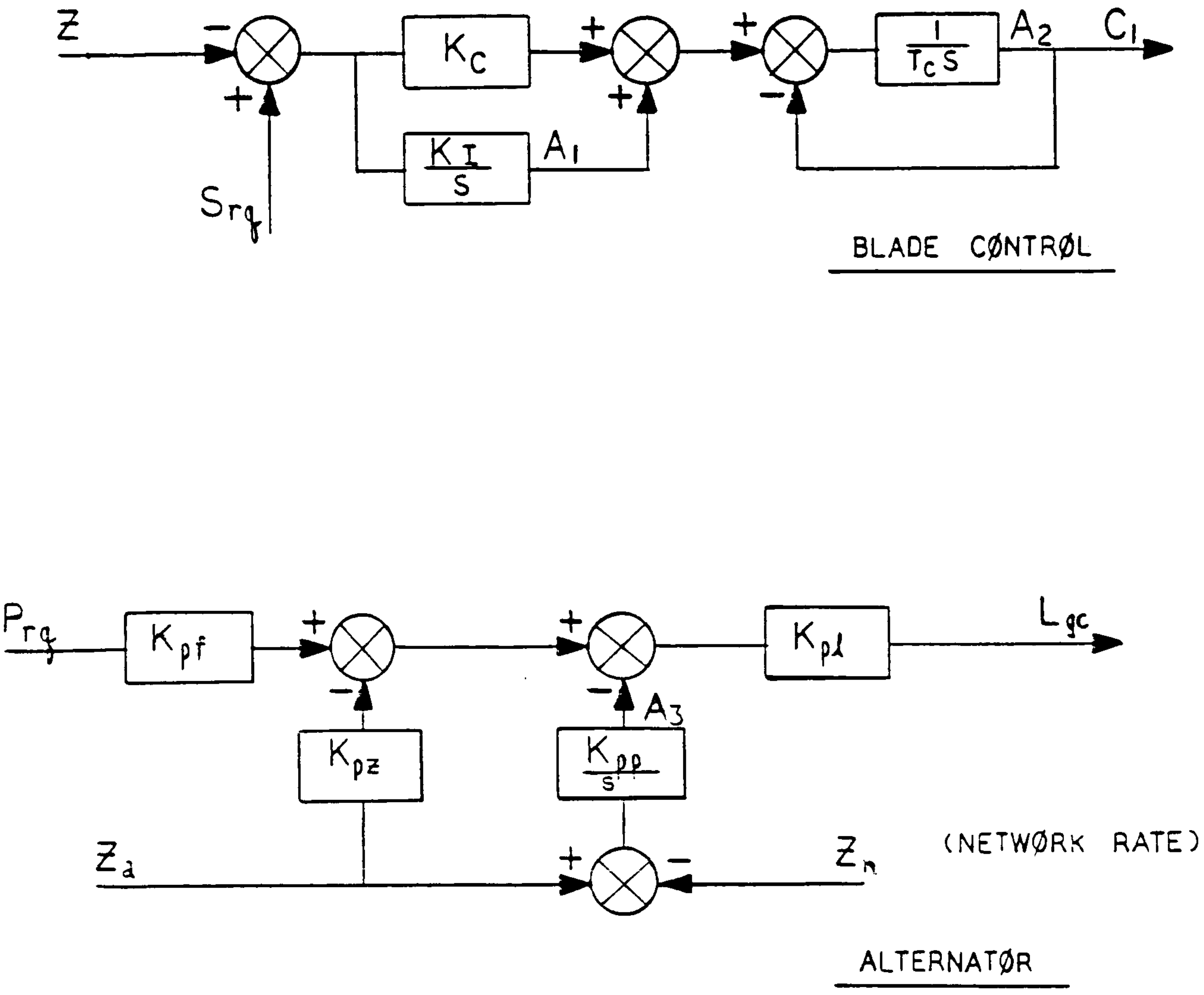


Figure 2.4. Example of a Typical Wind Turbine Control System

Referring to Figure 2.4, the equations for this system can be written as follows; the basic equations are shown first, followed by an equivalent version which conforms to the general state controller model.

Basic Equations:

Controller:

$$Z_e = (S_{rq} - Z)$$

$$A_1^{\bullet} = K_I \, Z_e$$

$$A_2^{\bullet} = (1/T_c) \, (A_1 - A_2 + K_c \, Z_e)$$

Alternator

$$A_3^{\bullet} = K_{pp} \, (Z_a - Z_n)$$

$$L_{gc} = K_{pl} \, (K_{pf} \, P_{rq} - K_{pz} \, Z_a - A_3)$$

Rearranged Equations:

$$Z_e = (S_{rq} - Z)$$

$$A_1^{\bullet} = A_{z1} \, Z_e$$

$$A_2^{\bullet} = A_{a21} \, A_1 - A_{a22} A_2 + A_{z2} Z_e$$

$$A_3^{\bullet} = A_{g3} \, Z_a + A_{0\dot{3}}$$

$$L_{gc} = L_{a13} \, A_3 + L_{pr} \, P_{rq} + L_g \, Z_a$$

$C_1 = C_{a12} \, A_2$ .... Blade Pitch Angle

Once the derivatives, $a^{\bullet}$, are computed, the control column, a, is obtained by numerical integration. The Euler integration method has been used in WEST-3; the time derivative is simply multiplied by the time step to get the change in a over a time interval. This method is suitable in most cases because the periods associated with most wind turbine control systems are long compared with the numerical integration step size. If the control loops incorporate high gains, and therefore produce high frequency control modes with short periods, then a more sophisticated numerical integration algorithm might be indicated.

2.4 Power Train

The following tasks are performed in the power train model:

* Compute the rotational acceleration and velocity of the rotor, taking into account the dynamics of the power train components.

* Determine power generated and the reaction loads applied to the support.

* Advance the azimuthal position of the rotor.

The general linear power train model in WEST-3 can be written as,

$$J_p \cdot p^{\cdot\cdot} + B_p \cdot p^{\cdot} + K_p \cdot p = p_m M_{h3} + p_L L_{gc} + p_{zd} Z^{\cdot}$$

$p \Rightarrow (p_1, p_2)$, power train variables; $Z = p_1^{\cdot}$; $Z^{\cdot} = p_1^{\cdot\cdot}$

A standard differential equation is incorporated, with inertia, damping and stiffness arrays, J_p.., B_p. and K_p.. Variable inputs to the model include the rotor torque, M_h, generator control torque, L_{gc} (i.e., torque applied to the generator rotor by the electrical system which is part of the control system in these WEST-3 models) and rotor spin acceleration, $Z^{\cdot}$.

Appendix E presents a derivation excerpted from Reference 4 which provides for a single power train degree of freedom. This simple analysis reveals the "algebraic loop" problem that rises naturally from the derivation of a power train model: this loop will drive almost any computer implementation of a power train model unstable. The reformulation used in Appendix E to solve this problem requires subtracting the influence of $Z^{\cdot}$ from the rotor shaft torque, and including this inertial effect instead in the power train equations. This approach, which is essential for successful power train simulation, requires the parameter, $Z^{\cdot}$, as an input. The generation of this parameter should be the same (in time) as that sent to the rotor to produce the rotor torque being used to solve the power train equations over a time step.

However, in some cases, even the reformulation outlined above may not prevent numerical instabilities when the power train model is coupled with the rotor model. In fact, the problem occured during the WEST-3 validation studies when the MOD-0 wind turbine was being simulated. Full details of the nature of the instability, and the special procedure devised to correct it can be found in Reference 7.

Figure 2.5 shows an example power train with two degrees of freedom.

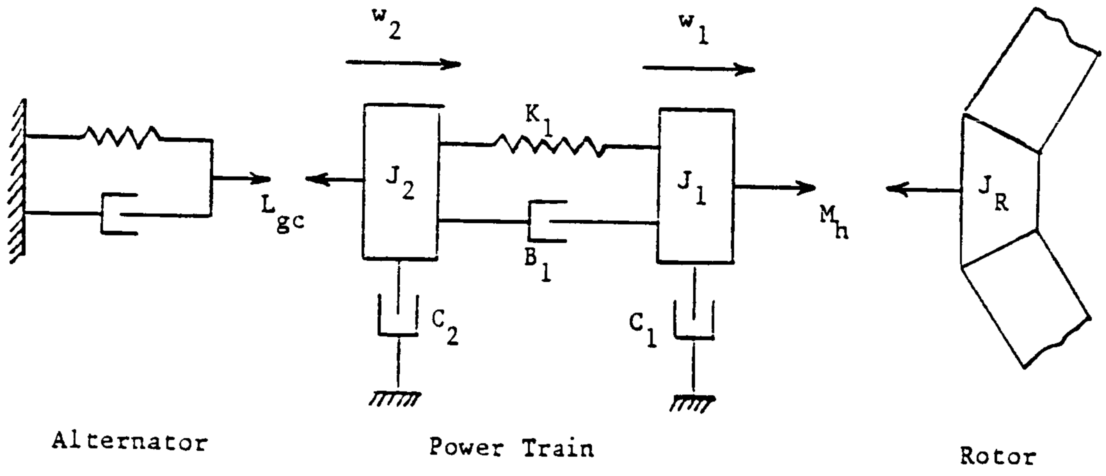


Figure 2.5 Example of a Power Train Model

Referring to Figure 2.5, the governing equations are,

$$J_1 \ddot{w}_1 = - K_1 (w_1 - w_2) - B_1 (\dot{w}_1 - \dot{w}_2) - C_1 \dot{w}_1 + M_h$$

$$J_2 \ddot{w}_2 = + K_1 (w_1 - w_2) + B_1 (\dot{w}_1 - \dot{w}_2) - C_2 \dot{w}_2 - L_{gc}$$

These basic equations have to be "reformulated" (see Section 3.1 for a diccussion of reformulation) to maintain computational accuracy. Accordingly, the generic positional coordinates are replaced by "differential" coordinates (which are the differences between positions of neighboring inertias in the model). The reformulated equations are,

$$\ddot{p}_1 = - K_{p12}\, p_2 - B_{p11}\, \dot{p}_1 - B_{p12}\, \dot{p}_2 + p_{m1} M_h + p_{zd1} \dot{Z}$$

$$\ddot{p}_2 = - K_{p22}\, p_2 + B_{p21}\, \dot{p}_1 - B_{p22}\, \dot{p}_2 + p_{m2} M_h + p_{zd2} \dot{Z} + p_{L2} L_{gc}$$

where,

$$p_1 = w_1 \qquad Z = \dot{p}_1$$

$$p_2 = w_1 - w_2 \qquad \dot{Z} = \ddot{p}_1$$

$Z_a = (\dot{p}_1 - \dot{p}_2)$ ... Alternator Speed

The constants in the equations are defined as,

$$J_T = J_1 + J_R$$

$$K_{p12} = K_1 / J_T$$

$$K_{p22} = (K_1/J_T) + (K_1/J_2)$$

$$B_{p11} = C_1 / J_T$$

$$B_{p12} = B_1 / J_T$$

$$B_{p21} = (C_2/J_2) - (C_1/J_T)$$

$$B_{p22} = (B_1/J_T) + (B_1/J_2) + C_2/J_2$$

$$p_{m1} = 1 / J_T$$

$$p_{m2} = 1 / J_T$$

$$p_{zd1} = J_R / J_T$$

$$p_{zd2} = J_R / J_T$$

$$p_{L2} = 1 / J_2$$

The reader is directed to Reference 5 for more detailed derivations of power train models. The Reference shows that even very general power distribution system models convert to expressions of the form presented in this Section.

The azimuthal position of the rotor is advanced by using a "rate resolver" algorithm. Actually, the sine and cosine of the azimuthal angle are generated as follows:

$$\sin(Y+DY) = \sin(Y)\ \cos(DY) + \cos(Y)\ \sin(DY)$$

$$\cos(Y+DY) = \cos(Y)\ \cos(DY) - \sin(Y)\ \sin(DY)$$

where Y is the current azimuthal angle, and DY is the advance angle which is a function of the numerical integration step size and the rotor speed. However, the algorithm is unstable in the sense that amplitude errors cause the sum of the squares of sine and cosine to be other than unity. To correct this problem, the cosine of the advance angle is varied small amounts from its nominal value (near 1.0) to correct for amplitude errors. The procedure is as follows:

Given: Z (rotor speed), DT (numerical integration step size),
current values of cos(DY), Sin(Y), and cos(Y).

i. Compute sin(DY): $\sin(DY) = Z\ DT$

ii. Compute the amplitude error: $e_a = \sin^2(Y) + \cos^2(Y) - 1$

iii. Make correction to cos(DY): $\cos(DY)\big|_{new} = \cos(DY) - G_a\ e_a$

iv. Compute the sine and cosine of the new azimuthal position, by using the new, corrected value of cos(DY).

Here the gain factor G_a has be chosen so that the algorithm is stable. For wind turbine simulation work the choice of $G_a = 0.25$ yielded good results.

2.5 Air Mass

The following tasks are performed in the Air Mass model:

- Define the linear and rotational velocities of air flow around the rotor.

- Compute the air flow retardation velocity as a function of the rotor thrust by using the standard Glauert momentum model (Reference 6).

- Simulate the effects of wind gusts by generating random number functions which appear as white noise in the system. The desired gust spectra are obtained by using quadratic filters.

- Determine the influence of wind shear and tower shadow on the air flow as seen by the blades.

The equation for the retardation velocity can be written as,

$$V_{rw} = K_{ar}\ RHO_r\ F_{h3} / (V_{ah2}^2 + V_{ah3}^2)^{1/2}$$

where V_{rw} is the retardation velocity, F_{h3} is the rotor thrust, RHO_r is the air density ratio, K_{ar} is a constant, and V_{ah2}, V_{ah3} are the local aerodynamic velocity components. Note that an implicit loop exists in the model, since the local aerodynamic velocity is itself a function of the retardation velocity. The problem is solved by a servo loop shown in Figure 2.6.

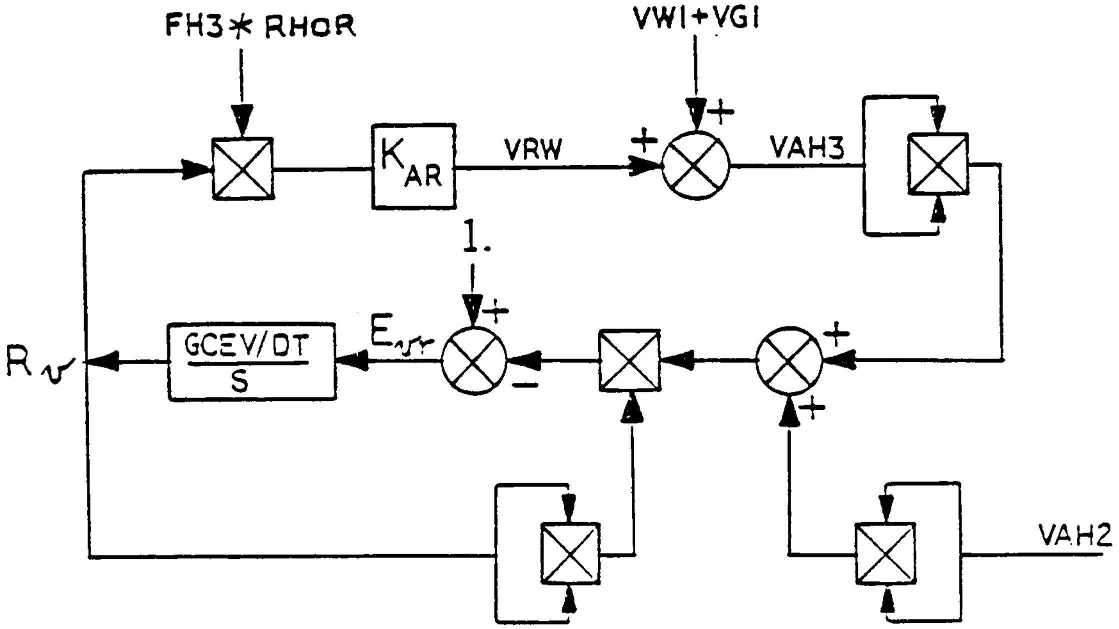


Figure 2.6 Servo Loop for the Air Flow Retardation Velocity

The servo loop solves for the reciprocal of the local aerodynamic velocity,

$$R_v = 1./(V_{ah2}^2 + V_{ah3}^2)^{1/2}$$

The calculation first computes an error function,

$$e_{vr} = 1. - R_v^2\ (V_{ah2}^2 + V_{ah3}^2)$$

which is zero if R_v is correct. This error is multiplied by a gain factor, G_{cev}, (which is input data to WEST-3) to produce a change in R_v which is in the direction to correct the error.

The size of the loop gain, G_{cev} will determine the speed at which the servo will converge on the correct solution. Excessive gains will lead to instability, however. Because the servo is nonlinear, the stability margin for the system varies over the operational envelope.

The local aerodynamic velocity can now be written as,

$$v_{wt} = v_w + i_1 V_{rw} + v_g$$

$$w_{wt} = w_w + w_g$$

$$v_{ah} = v_h + R_2 (Y_{r2}) . v_{wt}$$

$$w_{ah} = w_h + R_2 (Y_{r2}) . w_{wt}$$

The retardation velocity computed by the servo discussed above is subtracted from the ambient wind. The steady windspeed is specified by the two input columns, v_w and w_w. To these are added gust columns, v_g and w_g.

Windgusts are simulated by generating random number functions which appear as white noise in the system, and then by passing these through quadratic filters to produce gust spectra with controllable bandwidth, magnitude and center frequency. Six independent filters are incorporated, so that the wind can gust in six degrees of freedom near the rotor. The equations are,

$$\ddot{q}_g + B_g . \dot{q}_g + K_g . q_g = r_g$$

$$v_g = V_{gu} . \dot{q}_g$$

$$w_g = W_{gu} . \dot{q}_g$$

The windshear modulates the windspeed locally at a blade element depending on the distance the element lies above the ground. A table look-up scheme permits the use of an arbitrary nonlinear profile for the wind. The tower shadow effect (windspeed changes due to proximity of a blade to the tower) is usually specified as a percentage by which the wind speed is reduced, when the blade is within the shadow region, either rectangle or a sector. In general, windshear and tower shadow effects on the local blade aerodynamic velocity are implemented by calling two subroutines as follows:

$$S_{hr} = \text{SHEAR} \ (\cos (Y_i), \sin (Y_i), S_j)$$

$$S_{hd} = \text{SHADOW} \ (\cos (Y_i), \sin (Y_i), S_j)$$

$$v_{aa} = S_{hr} * S_{hd} * v_{ah}$$

where Y_i is the azimuthal angle of the i-th blade, S_j is the position j-th radial station, S_{hr} is the correction factor due to wind shear, and S_{hd} is the correction factor due to tower shadow. The augumentive velocity, v_{aa}, is added to the local aerodynamic velocity to account for the wind shear and tower shadow effects. In some cases, such as when the wind shear profile is linear, the corrections to the local aerodynamic velocity can be obtained by simple algebraic equations, instead of calling subroutines.

3. SIMULATION MODEL: REFORMULATION, NORMALIZATION, AND SCALING

The simulation of a wind turbine system begins with the mathematical modelling of the various physical phenomena which are occuring in the system. The mathematical models are then "processed", taking cognizance of any special numerical problems, and the characteristics of the simulator on which the models are to be solved. For example, all the computations in WEST-3 are done in 16-bit fixed point arithmetic with the attendent limitations in accuracy of the computations. Special software could be written to perform 16- or 32-bit software floating point operations. However, such software would drastically reduce the speed of simulation. Therefore, 16-bit fixed point computations have been retained in the WEST-3, and the simulation model has to take cognizance of this fact.

The objective of processing the mathematical models is to come up with a simulation model which permits

1. Accurate simulations in real time
2. Adapation to wind turbines of varying physical sizes.

Presented in this section are details of processing needed for wind turbine simulation on WEST-3. There are essentially three steps; Reformulation, Normalization, and Scaling.

3.1 Reformulation

Many subtle numerical problems arise in a wind turbine simulation; in particular, in the solution of the aeroelastic rotor equations. There are effects in the model that produce very large numbers. In the final results, differences of these large numbers yield small numbers which reflect some very important dynamical characteristics. The computation of the small differences of large numbers can lead to inaccurate results even in computers having large word sizes. On WEST-3, with its 16-bit fixed point arithmetic, this problem can completely destroy the accuracy of the simulation.

The answer to the problem is to reformulate the equations in such a way that small, important effects are not masked during the computations. The idea is to generate equations where subtractions of large numbers have already taken place so that, in the simulation, one solves directly for the important small-difference terms. The specific ways of reformulation must be selected with a thorough knowledge of the physics of the problem. The models and their special idiosyncracies must be well understood so that the critical areas that produce numerical problems can be identified and reformulated. Full details of the reformulation of the aeroelastic model used in WEST-3 can be found in Appendix B.

3.2 Normalization

The reformulated model goes through a process of normalization. This is a process of redefining all the variables in the model by using characteristic or reference values of dynamical quantities, such as, velocity, acceleration, force and moment. The normalized parameter will usually exhibit the same range of excursion, whether the machine being simulated is large or small. Though not a requirement, the process yields, in most cases, nondimensionalized equations. The objective of normalization is keep each number in the simulation within a reasonable range. The same normalized simulation model, with perhaps minor modifications, can be used for wind turbines of widely varying physical sizes.

Each entity (variable/constant) in the simulation has a normalization factor defined as follows:

$$E_s = E_p * F_s * F_n$$

where E_s is the "system entity" or the simulation entity, and E_p is the "physical entity" or the actual physical entity. In the current WEST-3 implementation, units on physical entities are feet, pounds, seconds, and radians, unless otherwise specified. Degrees, inches or metric units are not incorporated with physical entities. The "scale factor", F_s is described in the next subsection, 3.3.

F_n is the "normalization factor". For example, if E_p is the rotor thrust, then F_n will be the reciprocal of some force (e.g., the average thrust or the mean value of the design thrust that can be developed), so that $E_p * F_n$ is nondimensional.

In the definitions of variables and constants in this section, the normalization factors are expressed as ratios. The absence of a slash in these definitions is interpreted as a default numerator.

The normalization factor definitions thus appear in the forms

```
F(numerator)/F(denominator)
```

where the numerator and denominator factors can be FORTRAN expressions. In these normalization factors, the numerators and denominators are viewed separately. Only one slash is allowed within each definition. Examples of normalization factors are:

```
TH/RF          NB*PXA*C/RHO*R*R*W          1./RHO*V*V
```

Note that these factors are <u>not</u> decoded as a standard FORTRAN expression. For example, standard FORTRAN decoding begins at the left of an expression and performs the operations as it moves to the right. FORTRAN would decode the second normalization factor, above, as follows:

```
(NB*PXA*C/RHO)*R*R*W
```

In the conventions defined for normalization factors herein, the factors are considered pure ratios. Proper decoding for this convention in the example above would therefore be:

```
(NB*PXA*C)/(RHO*R*R*W)
```

3.3 Scaling

After normalization, the simulation model is "scaled", a process which ensures that each variable and constant in the model takes on values within the range of +1 and -1. Scaling is needed because all the computations in WEST-3 are done in 16-bit fixed point arithmetic, a feature which is a major contributor toward achieving a real-time simulation capability. The limited significant figure accuracy attendent in such arithmetic demands that special scaling techniques be used to process the equations (and the constant data) before they are solved on WEST-3. The scaling process is similar to the one used in analog computers. However, unlike analog computers, the all digital WEST-3 is rather unforgiving of scaling errors which cause over/underflows.

In conjunction with the normalization factor, the scale factor relates a physical entity to the corresponding system entity, i. e.,

$$E_s = E_p * F_s * F_n$$

The scale factors, F_s, are numbers. They are selected so that the system entity does not exceed the range $-1 < E_s < +1$. In the simulation, the system entities are 16-bit numbers that can be viewed as having a decimal point to the far left of the number. In the Computational Units (CUs) of WEST-3, the numbers are represented in WEST-3 in 2's complement form. Hence, these numbers can only range between +/-.9999 (decimal). The CUs incorporate a hardware multiplier that performs 2's complement multiplies assuming the decimal point placed as defined above. Hence, two numbers with maximum values multiply to a number also at the maximum value incorporated in the system.

3.4 Database Representations of Variables and Constants

Complete definitions of all the variables and constants in the simulation model are given in Appendix F. Each definition consists of two lines. The first line shows:

Label(s): Scale Factors : Normalization Factors

and the second line of each definition is an arbitrary field of 80 characters expressing the definition of the entity in English.

The labels are the symbolic names of the entities that appear in both the physical and scaled equations. The label field is terminated by a colon. Scale factors follow as a string of numbers separated by commas. Normalization factors follow a colon. The normalization factors are separated by commas. In these definitions, blanks have no significance, and can be placed anywhere to aid in the clarity of the definitions.

The syntax associated with variable and constant definitions uses an "autospawn" capability: a shorthand procedure whereby many definitions can appear on any line. The autospawn facility operates on an entity enclosed by commas, a colon and comma, or a comma and the end of the entire line. The autospawn syntax, in general, has the form:

CCCC(i)CCC+n

where the "C"s are characters of essentially any definition except parentheses or the + or - sign. The autospawn syntax defines n+1 items. The first appears if the parentheses and the "+n" are dropped. The next is with i+1 substituted for i, and the parentheses and "+n" dropped. The last defined entity is with the numeral "n" substituted for the "i".

For example, the syntax:

```
VA(3)+4
```

defines a string of labels:

```
VA3,VA4,VA5,VA6,VA7
```

In some cases, the autospawn facility is used without parentheses to repeat an identical definition. For example, the syntax:

```
.3333+3
```

Produces the equivalent of a string of four numbers ".3333". With this definition, consider the following example of a variable definition using autospawn:

```
VW(1)+5:     2.5+5:     1./VR+2,     1./OR+2
WIND INERTIAL VELOCITY RESOLVED TO OVERALL SYSTEM COORDINATES.
```

This definition defines six scalar entities, VW1, VW2 ....VW6. All have scale factors of 2.5. The first three in the string have normalization factors 1./VR (where "VR" in this case is a velocity reference = rotor tip speed in the case of the rotor), and the final three entities in the string are normalized to the reciprocal of "OR" (Omega reference- a rotational velocity reference which in the case of the rotor is the nominal rated rotor spin rate).

The autospawn facility is very powerful, especially for systems which tend to be organized in vector strings. The size of the data base required to define a complex set of system equations is considerably reduced using this autospawn definition. The autospawn can produce decreasing numbers in a definition if a minus (-) sign is included in the parentheses. For example, a definition of the form

```
VAB(-6)+5
```

might be convenient, where the autospawn would produce the equivalent definition:

```
VAB6, VAB5, VAB4,....VAB1.
```

Note that the final number in the string can be calculated by subtracting the number following the "+" sign from the number in parentheses. The similar situation emerges for ascending autospawn definitions, where the last number in a string is the sum of the one in the parentheses and the one after the "+" sign.

3.5 Physical Equations and System Equations

The math models for the simulation are presented in this report section in two forms: "physical equations" and "scaled equations". The physical equations are direct representations of the reformulated mathematical models, expanded to scalàr form. These equations incorporate physical variables and constants. The scaled equations are those actually solved by the computer.

Appendix G presents the physical equations for the wind turbine simulation model consisting of the Rotor, Air Mass, Tower, Power train, and control system. The corresponding system equations can be found in Appendix H.

The parameters in the scaled equations are the system parameters. The equations appear very much like the physical equations, except that many terms in them have additional factors. These factors are always real numbers. The scale factors are chosen so that these numbers are usually bounded between +/-1, just like system parameters. The numbers can be out of this range if they are integral powers of 2, however.

The factors in the scaled equations have been placed there so that the scaled equations are equivalent to the physical equations. To see how this is required, suppose the following normalization and scale factors have been defined for certain variables and constants:

Symbol	Scale Factor	Normalization Factor
X	5.	S/W*R
Y	2.	T/P*Q
Z	20.	R/W
A	6.	R*R/S
B	25.	P*P*Q*Q*R/T*T*W

Let the physical equation appear as follows:

$$Z = A * X + B * Y * Y$$

In this example, the scaled equation (i.e., the system equation) will be

$$Z = .6667 * A * X + .2 * B * Y * Y$$

One can confirm the equivalence of the system equation to the physical one by substituting physical parameters (with the normalization and scale factors included) in lieu of the corresponding system parameters in the system equation. The items in the normalization factors should all cancel and all real constants should merge into unity factors, such that the system equation so processed becomes identical to the physical equation.

As stated before, the factors that appear in the system equations (placed there for equivalence to the physical equations) are usually bounded between +/-1. Indeed, the scale factors on the variables and constants are chosen so that this is the case. When this is true, the hardware multiplier is used to simply multiply the number by the other factor making up a full term in an equation. If this number appears as a product with another real number in a term, the numbers can be combined into a single number. For example, if a term in a physical equation contained a number "2.", and a factor (arising from the scaling considerations) of ".3333" is also needed on the term, the final numeric factor would be ".6666", assuming of course that this final factor obeys the rules.

The rules for factors are: bounded by +/-1 or integral powers of 2. In the first case, the factor is processed in a normal way. If the factor is represented as a power of 2, it must be considered a shift operation in the code, not a multiply. For example, a factor of 16. would require a number to be shifted four times to the left. An arithmetic shift would be made, so that the bits that appear in the four rightmost positions in the word (after shifting) are zero.

Division by integral powers of 2 can also be accomplished by shifting right. Hence, it is correct to show divisions in equations, but this is seldom of value since the division by 2 can be easily represented as a factor of ".5".

It should be noted that one shift takes the same time in a CU as a multiply. Factors larger than 2 will require more that one instruction, however, so they are slower than multiplies and should therefore be avoided unless absolutely necessary.

Scale factors and normalization factors are chosen based on a sound understanding of the physics associated with the system being simulated. After these factors are initially chosen, they are often changed many times as a simulation is brought to operational status. The changes occur because of many observations, including:

* A constant, after scaling and normalization, exceeds the range +/-1.

* During simulation runs, a system variable is found to exceed its range of +/- 1 for legitimate operating conditions of interest.

* A numerical factor in a system equation is not within the rules (within +/-1) or an integer power of 2.

* Inaccurate simulation results occur because important terms in the equations are masked in the final calculations due to poor scaling (leading to excessive loss of significant figure accuracy).

4. DYNAMIC SCALING

Most of the scaling associated with the simulation equations incorporates "fixed" or constant scale factors; these have already been discussed in Section 3. In key areas, however, variable scale factors are needed because of the propensity of certain expressions to have severe ranges of operation that preclude accurate simulation with only 16 bits and a constant scaling.

In these cases, variable scale factors are used. These are in every way equivalent to floating point exponents. These are only used where they are definitely needed because they involve more operations (and therefore take more time) than simple fixed-point operations.

When variable scale factors are used, they are interpreted as positive binary exponents. An entity will thus be represented as

F * 2**SF

where F is the fraction and SF is the variable scale factor. The fixed scaling for the entity can be chosen so that SF is always a positive integer number. Hence, SF is implemented in the programming by loading F into the accumulator, multiplying it by the factor associated in the expression and then shifting the result SF times to the left in the accumulator. The undefined least significant bits (SF of them) in the word are set to zero in this process.

In most cases where variable scale factors are used, two computer words are used, one for F and the other for SF. In some cases, however (e.g., large tables), both F and SF can be packed into one word. For example, of SF is given 3 bits, then one 16-bit word can carry the sign bit, a 12-bit fraction (which is an accuracy of +/-.025%) and an exponent that can expand the range of the parameter to +/-256 from the usual range of +/-1.

The 16-bit floating point format described above is very accurate in representations of, for example, aerodynamic tables where the physical data in the tables is usually not known to within 5 per cent.

4.1 Aerodynamic Geometry

This subsystem involves a special problem: the calculation of quotients that can have a small denominator. The subsystem resides in the rotor blade element computations, the group of computations in the simulation that are most speed critical. For these reasons, special considerations have been made regarding these computations.

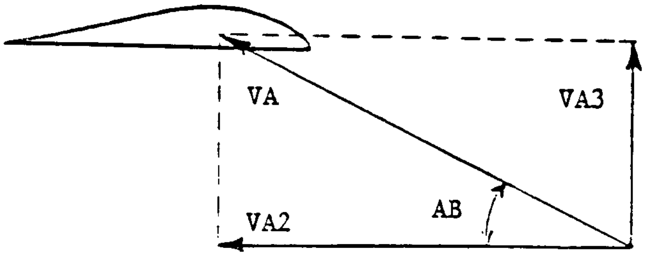
The computations essentially require division, a process that is not incorporated in the hardware of the Computational Units (CUs) of WEST-3. Software divisions can be done, of course, but these are slow and therefore very undesirable in this most critical area of the simulation.

Other considerations on the aero geometry calculations are-

* the results go to the aero tables- tables of physical data that are never known to high degrees of accuracy. Because of this, the results of the aero geometry need not be any more accurate than the tables are known. The aero geometry calculation must be made with speed and accuracy carefully balanced so that excessive (unneeded) accuracy does not penalize speed.

* The calculations are made in an integration loop, so that the values of parameters entering and leaving the calculations change only a small amount for every pass through the calculations. This fact allows the use of algorithms that take advantage of the perturbation nature of the process, using past values as starting points for new calculations.

Figure 4.1 shows the aerodynamic geometry at blade radial station.



```
VASQ = VA2*2 + VA3**2

VA   = SQRT( VASQ )

SAB  = VA3 / VA

CAB  = VA2 / VA

M    = RVC * VA
```

Figure 4.1 Aerodynamic Geometry at the Blade Station

Here VA is the air speed, VA2 and VA3 are its components, SAB and CAB are, respectively, the sine and cosine of the angle of attack AB, M is the Mach number, and RVC is the reciprocal of the local speed of sound. The Mach number calculation is straightforward and requires no additional discussion here. The others involve the square root and division operations, however, both problematic in the simulation.

The calculation of airspeed is easily performed if the changes in SAB and CAB are known. If the expression for the square of the airspeed is differentiated and then divided through by 2*VA, the change in airspeed is given as:

```
d(VA) = CAB * d(VA2) + SAB * d(VA3)
```

Hence, if the airspeed was known for the last pass through the equations, the old values of SAB and CAB can be used with known differentials on the input velocity components VA2 and VA3 to get the perturbation in airspeed. A new estimate of airspeed can then be made, and used in the calculations of new SAB and CAB parameters.

The real problem is the division by airspeed. An algorithm will now be derived that converges to the reciprocal for small changes in the input parameter.

Let V be the known parameter, where it is necessary to compute $x = 1/V$. If x_o is the last known value of the reciprocal, an error function, e, can be defined using the new (given) V:

$$e = (V * x - 1.)$$

If x is correct, e vanishes. The calculated error for x_o is

$$e_o = (V * x_o - 1.)$$

The derivative of e with respect to x is V. Since e needs to vanish, the equation for dx is derived:

$$e_o + de = 0 = e_o + V * dx$$

Solving for dx and noting that $dx = x - x_o$, where x is the new desired result:

$$x = x_o - e_o/V$$

Approximating 1/V with the last value of x, x_o, the correction expression is

$$x = x_o * (1. - e_o)$$

This is the final expression, since it is desirable to compute e_o as an intermediate calculation to see how close the convergence is. The e_o can be eliminated, however, producing another version of the algorithm:

$$x = x_o * (2. - V * x_o)$$

If V changes by small intervals, this expression converges rapidly to the desired result using minimal calculations. Additionally, the error can be tested and the number of iterations expanded. in cases where the error is too large due to rapidly changing V.

This algorithm can be used to compute the reciprocal of air speed, RECVA, which can be carried in the aero geometry section to facilitate straightforward calculation of SAB and CAB. The problem with this is that the reciprocal can span a very large range because airspeed can get very small.

Variable scaling can be used to solve this problem where RECVA is carried as a fraction and integer scale factor (floating point number). The logic in the calculations detects overflow when RECVA is computed, which calls for incrementing the scale factor and shifting the fraction right by one (to eliminate the overflow). The scale factor can be decremented and the fraction shifted left if the most significant bit of the fraction is detected as a zero.

The floating point operations described above are simplified by the fact that RECVA is always a positive number.

4.2 Aerodynamic Coefficient Tables

The aerodynamic tables (for the wind turbine application where Mach number is irrelevant) use SAB and CAB to produce lift and drag coefficients CFA3 and CFA2. The tables must apply to a 360-degree range of angle of attack, AB, but they are both univariate tables in that only one of the two input parameters are used to look up within the tables at a time.

The tables are separated into four quadrants defined by the points where the sine and cosine of ABI are +/-.707. The quadrants and the associated independent variables ("lookup parameters") are described in the table, below:

Quadrant	Definition	Lookup Parameter
1	CAB > .7071	SAB
2	SAB > .7071	CAB
3	CAB < -.7071	SAB
4	SAB < -.7071	CAB

Each of the quadrants has two tables, one for CFA3 and one for CFA2. These tables have a number of elements that must be an integral power of 2, but the numbers of elements in the tables associated with each quadrant can be different. For example, quadrant 1 may have 1024 entries for CFA3 and an equal number of CFA2 points. Quadrant 2 may have only 128 elements in each table, however.

Parameters are fetched from each table simply by using the lookup parameter as an address. The lookup parameter is masked to the correct number of bits by multiplying it by a mask word. The result is added to an offset, and the indirect address so created is used to fetch the desired CFA3 and CFA2 numbers. By providing adequate granulation in the tables, interpolation can be avided.

Because of the ranges that the aero coefficients can reach (particularly CFA2), the tables are be 16 bit floating point numbers (three bit binary exponent). The fixed scaling will enable the use of only positive integer exponents.

The floating point numbers fetched from the tables are decomposed into separate fraction and exponent words for use in computing the Aero Loads at a blade radial station; products of these numbers with the dynamic pressure and the blade chord will be shifted by the number of places of the exponent to produce the final distributed blade aerodynamic loads, lift and drag.

5. CONSTANT DATA PREPROCESSOR

In addition to the gereration of the simulation model, described in the previous sections, the other major task associated with a wind turbine simulation is the preparation of the constant data. The data consists of physical data and specifications of the specific wind turbine to be simulated; the rotor dimensions, the wind speed, commanded rotor speed, the frequencies and mode shapes if the blade modes etc.. Additionally, there are a large number of constants which have be computed for use by the simulation model; for example, products involving the blade mass distribution and the elastic mode shapes integrated over the blade span.

All this raw data has to be normalized and scaled before it can be used in the simulation. In Figure 1.1 of Section 1, this prosess of generating the scaled constants is indicated as being carried out by a "PREPROCESSOR". In actuality, the preprocessor consists of more than one specially designed computer program, running on a computer external to WEST-3. At present, Digital Equipment Corp. (DEC) PDP11 computers are being used for this purpose at Paragon Pacific Inc.. This section presents a outline of the mechanics of generating the scaled constants.

Figure 5.1 shows a schematic of the computer programs and the various data files used in processing the constant data. The decision to have a number of programs, instead of a single large program, was made mainly for debugging purposes. It is envisaged that, for future work, a more compact version of preprocessor will be developed. Referring to Figure 5.1, there are two distinct components, generation of the physical constants, and the generation of an executable load module which can convert the physical constants into scaled constants. Following are brief descriptions of the components.

DSS2:

This program is used to process all the data associated with the wind turbine rotor. The input to the program is a file, DSS2.INP, which contains data pertinent to the rotor and the operating conditions. The major task of this program is to relieve the user from having to generate the the large number of constants needed for the simulation. The output file, DSS2.OUT, contains an echo of all the input data, and a listing of the generated constants. The other two output files, DSS2.SNG and DSS2.TBL, also contain the same information, for use by the WEST load module.

WEST.CON

This file contains all the constants pertaining to the subsystems, other than the rotor, i.e., Air Mass, Tower, Power Train, and Control System. WEST.CON is directly input into the load module, WEST.

CONVRT:

The task of this program is to convert the data base representation (see Section 3.4 and Appendix F) of the variables and constants used in the simulation into standard FORTRAN programs. By using the system software of the DEC PDP11 computer, these FORTRAN programs can then be compiled and linked to yield an executable load module, WEST.

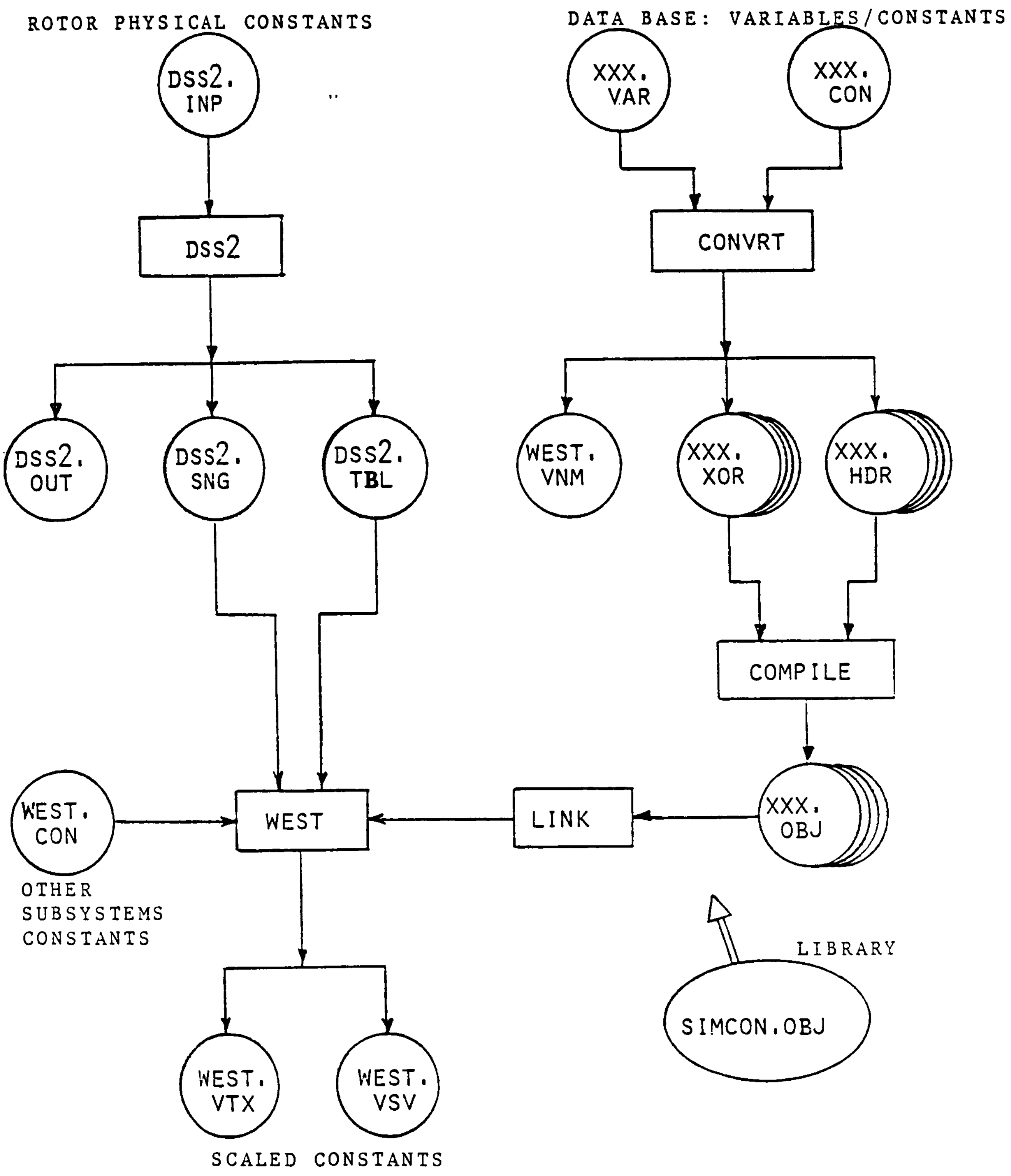


Figure 5.1. Generation of Scaled Constants: PREPROCESSOR

As an example, a typical data base representatation of a variable is,

KGE : 20.0 : MR / RAR
CONSTANT IN THE GIMBALL SERVO INTEGRATION FORMULA

where MR is the reference moment, and RAR is the reference linear acceleration; both are defined in the output file DSS2.SNG. Here, the scale factor is 20.0, and the normalization factor is (MR/RAR). The CONVRT program will convert the data base representation into a FORTRAN arithmetic expression,

KGE = 20.0 * MR / RAR

The input files XXX.VAR and XXX.CON contain, respectively, the definitions of the variables and constants in the simulation (see Appendix F). The output files XXX.XOR and XXX.HDR together constitute the FORTRAN program containing arithmetic expressions, such as the one shown above for the constant KGE. Also, there is a third output file, WEST.VNM, which contains a list of the variables and constants.

COMPILE and LINK:

These steps of compiling and linking, to get the executable load module, WEST, are straight forward uses of the DEC computer's system software.

WEST

This is the executable load module, which does the actual conversion of the physical constants in the files, DSS2.SNG, DSS2.TBL, and WEST.CON into a form which can be downloaded into WEST-3. The output file WEST.VTX is a listing of all the scaled constants, for the user's reference. The same information is contained in XXX.VSV, in a binary format, for efficient processing by the WEST-3 system software.

SIMCON.OBJ

This is library of several special purposes subroutines which have been coded for use by the programs, DSS2, CONVRT, and WEST.

6. WEST-3 SYSTEM SOFTWARE

After the simulation model and the scaled constants have been prepared, the next step in the simulation process consists of using the WEST-3 system software to generate the executable code which can downloaded into WEST-3. From the inception of the design of WEST-3, it was recognized that having adequate system software was the key to making the system friendly to the user, and hence increase the usefulness of the system. The system software that has been developed for WEST-3 is fairly standard, in the sense that, the process of generating the executable code is similar to that in any other computer. A schematic of the WEST-3 system software can be found on page 1-2 of this report in Figure 1.1 of Section 1. Presented in this Section are brief descriptions of the various components shown in Figure 1.1.

The scaled simulation model consists of a set equations which are coded in a subset of ANSI FORTRAN, the computer language used widely in engineering applications. At present, the subset consists of the following:

1. arithmetic expressions
2. logical IF statements
3. GO TO statements
4. CONTINUE statements,
5. COMMON block definitions
6. FUNCTION subprograms
7. SUBROUTINEs.

This subset of FORTRAN has been found to be adequate for application to wind turbine simulations. Due to the modular design of the system software, future enhancements to the Translator can be readily incorporated. It should be emphasized that the user is not required to code the simulation model in the machine's native assembly language; programming in the assembly language is a nontrivial process.

The Translator converts the scaled model into WEST-3 assembly language mneumonics. The Assembler converts the output of the Translator into an Object Code suitable for processing by the Linker. The role of the Linker is traditional, in that it generates an executable Load Module by combining several Object Code modules; typically, a main program and several subroutines. The Translator, the Assembler, and the Linker constitute the bulk of the system software for WEST-3. For debugging purposes, several files, such as memory maps, are also generated while processing through the software.

The Loader performs task of loading the memories of a Computational Unit: the instructions (Load Module) into the Insruction Memory (IRAM), and the scaled constants into the Processing Memory (PRAM). The Loader has been designed to be as user friendly as practicable. For example, the choice of which Computational Unit is to be loaded, the list of input/output variables, configuration of the serial ports are all definable by the user at the time of loading.

Several utility programs are also available for debugging. Among other things they provide for the display of internal buses in a Computational Unit, single stepping through the program, and peek/poke capabilities. These utilites are invaluable for detailed debugging, when needed.

7. CONCLUDING REMARKS

As part of the WEST-3 development process, validation of WEST-3 was successfully completed by conducting simulations of the MOD-0 wind turbine. The validation exercise served to prove the WEST-3 system, both hardware and software. The exercise was also invaluable in providing direction for enhancing the utility and user friendliness of WEST-3. As a result of the experience gained during the development of WEST-3, the following efforts are either underway, or planned for the near future:

1. The method of coupling the rotor model to the power train and the tower models needs to be reexamined; reformulation of the presently used procedure may be indicated.

2. A more compact version of the existing preprocessor programs used for preparing the constant data needs to be developed.

3. Extensive scaling of the simulation model, mandated by the fixed point nature of WEST-3, is a tedious and time consuming task. To eliminate the need for scaling, a 32-bit floating point system has been designed. Breadboard verification of a single Computational Unit is under progress, and a full system is expected to ready for use early in CY 1986.

4. To significantly improve the user friendliness of the system, development of a new suite of system software is underway for the floating point system. The user will be able to write programs in the ANSI FORTRAN-77 language; presently, only a subset of the language is permitted. Further, the user will have access to many more debugging tools than are available at present.

REFERENCES

1. Sridhar, S., "WEST-3 Wind Turbine Simulator Development Final Report, vol 1: Summary,", PPI-FID-300101, Paragon Pacific Inc., NASA CR-174981, DOE/NASA 0247-1, July 1985.

2. Hoffman, J. A., and Weisz, D. R., "WEST-3 Wind Turbine Simulator Development Final Report, vol 4: Hardware,", PPI-FID-400063, Paragon Pacific Inc., NASA CR-174984, DOE/NASA 0247-4, July 1985.

3. Hoffman, J. A., and Thoren, R. J., "Mathematical Models and Hybrid Program for the Wind Energy System Time-Domain (WEST) Simulator - Baseline WEST Unit (Unit 1)," PPI-FID- 300072, Paragon Pacific Inc., Mar. 1978, Revision 6, Feb. 1979.

4. Hoffman, J. A., "Math Modelling and System Design Report: Conversion of the Wind Energy System Time-Domain Simulator, Unit 1 (WEST 1) System Design to the WEST 2 Design," PPI-FID-300070, Paragon Pacific Inc., July 1979.

5. Hoffman, J. A., Williamson, D. R., and Henninger, W. C., "Mathematical Methods Incorporated in the Wind Energy System Coupled Dynamics Analysis," PPI-FID-300015/16/17, Paragon Pacific Inc., Aug. 1977, Three Parts.

6. Gessow, A., and Myers, G. C., "Aerodynamics of the Helicopter," The MacMillan Co., New York, 1952, Chapter 3.

7. Sridhar, S. "WEST-3 Wind Turbine Simulator Development Final Report, vol 2: Verification,", PPI-FID-300102, Paragon Pacific Inc., NASA CR-174982, DOE/NASA 0247-2, July 1985.

8. Hoffman, J. A., "Test and Development Report: A Gimbal Subsystem For The Wind Energy System Time-domain (WEST) Analyzer," PPI-FID-300033, Paragon Pacific Inc., Dec 1979.

APPENDIX A

NOMENCLATURE, COORDINATE SYSTEMS, AND TRANSFORMATIONS

Page

This appendix is provided to clarify terms and define fundamental methodology incorporated in the analyses presented throughout this document. It is not intended to be comprehensive treatise on the methods of vector and matrix analysis. Portions of the appendix are tuitorial in nature to serve the purposes of definition and clarification, but a basic knowledge in the areas of vector and matrix algebra and calculus on the part of the reader is assumed.

A comprehensive analysis often involves many systems and system components which require the general methods of mechanics. Motion entities must be carefully defined and processed, and many transformations are required. Hence, it is the purpose of this appendix to define uniform conventions for these processes. Included in such definitions are:

* Definition of standard terms.

* Rigorous definition of coordinate systems, including shorthand acronyms for each to simplify categorization and documentation.

* Derivation of standard coordinate-transformation procedures and nomenclature.

* Definition of shorthand notations for use in rigorous definition of the various motion entities.

A.1 Definitions of Terms

This subsection serves to clarify the various terms applied throughout an analysis, particularly as regards the definition of motion entities.

A motion entity is a position, velocity or acceleration. Motion entities are generally expressed in either vector or vector column (matrix) form.

A.1.1 Vectors

A vector is used in vector mechanics to relate the positions or motions of two points. The vector can be translational (e.g., a position vector, or translational velocity) or rotational. A vector requires three specific definitions. To avoid confusion, one should specify these three definitions with checklist reliability:

1.--What type of vector is it (position, velocity, acceleration, translational, rotational)? This will be called the type (T) of vector.

2.--What is being located, or what is moving (point, axis system, etc.)? Call this item the subject (S).

3.--With respect to what (axis system, point, etc.) is the subject located or moving? Call this base (B).

For example, if one is defining the velocity vector V*, the checklist would require language such as

> "V* is the translational velocity [1. type] of aircraft c.g. axes [2. subject] with respect to inertial space [3. base]."

Vectors are rather abstract in that they do not require Resolution (R) for their definitions; i.e., a specific coordinate system need not be defined in order to rigorously define a vector entity. Vectors are simply "arrows" in space between point pairs, and represent whatever the analyst wishes to assign them.

A column vector or column matrix is numerical representation of a vector. The column vector is a group of numbers called "elements"; each represents a component of a vector resolved to a specific coordinate system.

Hence, the column vector requires a fourth item in the definition checklist, which is

4.--With respect to what axis system is the entity Resolved (R)?

As an example, let v be a column vector representing the vectorial entity V*, discussed previously. Then a proper definition of v would be

> "v is the translational velocity [1. type] of aircraft c.g. axes [2. subject] with respect to inertial space [3. base], resolved to c.g. axes [4. resolution]."

If any of these four key definitional items is not clearly indicated, then the column vector is essentially undefined and much confusion and error can result. This particularly is true when computer algorithms are used to solve for vector entities. It is the author's experience that poor definitions of motion entities and coordinate systems are the primary causes of errors in computer dynamic analyses.

A.1.2 Time Differentiation

As is well stated in standard texts on mechanics, careful definition of time differentiation is required. When the derivative of a motion entity is taken with respect to time, the specifc coordinate system in which the derivative is taken must be specified. This, of course, is because the time derivation of motion entity is different among coordinate systems which accelerate with respect to each other.

Derivatives of vectors are therefore seen to require a fifth definition: the axis system with respect to which the differentiation is taken. Call this the differentiation base (D) for the derivative vector.

A.1.3 Scalars

Vectors have directorial properties as described above, but scalars do not. A vector is represented by a group of numbers whereas a scalar is represented as a single number.

The mass of a body, for example, is a scalar entity. The velocity of the body would be represented by a vector. The magnitude of the vector is a scalar.

A vector can be expressed as the vector sum of three orthogonal vectors. The orthogonal vectors are parallel to the axes of some defined orthogonal coordinate system. The magnitude of the orthogonal vectors are scalars. These scalars are stacked in a column to become the "elements" of the matrix or "column-vector" representation of the vector.

The matrix representation of a vector -a group of scalars- is a very convenient vehicle for use in computer implementation of vector processes.

A.1.4 Operators

Rectangular arrays of scalars can be defined as "operators". These operators can be added and multiplied in accordance with specific rules defined for such operations.

In vector analyses, such operators are often called "tensors" or "second-order tensors". They arise naturally in the derivation of the rotational equations of motion for rigid bodies; in this form they are often called "inertia tensors". Another term sometimes given to the second-order tensor is the "diadic product".

Square matrix operators are also used for various transformation procedures. Such transformations are the subject of later sections of this document.

A.1.5 Nomenclature Conventions

Conventions are defined below which can be used to derive and document dynamics analysis equations. The conventions have been defined for convenient incorporation into computerized systems including word processors.

In order to use computer systems to store and print dynamical equations, some ground rules are needed which are somewhat different from expressions. The ground rules of this type used herein are:

* Greek or other nonstandard characters are not used.

* Groups of standard graphics characters are used to form some special symbols (e.g., the integral sign).

Equations can be formed on a computer-driven printer if super-scripting, sub-scripting and backspacing are allowed, and if the printer can process all characters in a full ASCII set. Many printers are available that can do this. Table A-1 presents the full ASCII character definitions.

Table A.1 ASCII Character Code Definitions

Bits b7 b6 b5 →					0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b4	b3	b2	b1	Column → / Row ↓	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	A	LF	SUB	•	:	J	Z	j	z
1	0	1	1	B	VT	ESC	+	;	K	[	k	{
1	1	0	0	C	FF	FS	.	<	L	\	l	|
1	1	0	1	D	CR	GS	-	=	M	]	m	}
1	1	1	0	E	SO	RS	.	>	N	^	n	~
1	1	1	1	F	SI	US	/	?	O	-	o	DEL

HEXIDECIMAL = ASCII COLUMN + ROW

With these ground rules already stated as a guide, the following specific nomenclature definitions can be made:

* Scalar entities are represented by groups of characters beginning with an upper-case letter. These are sometimes called "zeroth order tensors".

  Examples of scalar nomenclature:

  S_b , $\dot{V}$, M_b , Mq^{-1} , W(S)

* Column vector entities are represented by groups of characters beginning with a lower-case letter. These are sometimes called "first order tensors".

  Examples of column vectors are:

  v_b , w_a , $\dot{b}$, u(S)

  Note the functional dependency on the scalar parameter, S, in the last example, above.

* Second-order matrices can be viewed as operators on vectors or operators on the other second-order matrices. These are often called "tensors" or "second-order tensors" in dynamic analyses. Such entities are represented by groups of characters beginning with an upper case letter and ending with a period. The period distinguishes the operator from a scalar.

  Examples of operators are:

  R. , X_3. , F_{dv}. , X(v).

  Note the functional dependence of the X. operation on the vector v in the last example, above.

* Time derivatives of entities can be represented in a number of ways. Superscript dots and primes are often used to denote differentiation with respect to time and space (i.e., some nontime parameter), respectively. Differential notation is also sometimes used, e.g.,

  d (v) / dT

  for differentiation of the item in parentheses with respect to the scalar, T.

  In the present conventions, the superscript dot is used to represent time differentiation with respect to the axis system of resolution. For example, if v is a velocity resolved to, say, coordinate system RAX, then $v^{\bullet}$ is the time derivative of v taken with respect to RAX.

  Not withstanding this convenient nomenclature, clear and specific "checklist" definitions should be given in all derivations defining the differentiation base.

- The symbol for integration is

  $$\int_a^b I \, dS \ ,$$

  for integration of the interand I with respect to S between boundaries a and b.

  The summation process is indicated by, for example:

  $$\sum_i^N S_i$$

  where the scalars S_i are summed for i = 1,2...N.

- If a vector is to be defined from a group of scalars, it is convenient in printed material to show the column vector as a transposed row vector. For example:

  $$v = (V_1, V_2, V_3)^T$$

  defines a column vector as composed of three scalar elements V_1, V_2 and V_3 . Superscript T denotes the transpose of the row vector to produce the column vector.

- Identity vectors and operators are often needed in an analysis. The following conventions are therefore defined, for convenience.

  identity vector:

  $$i = (1, 1, 1)^T$$

  specific identity vectors:

  $$i_1 = (1, 0, 0)^T$$

  $$i_2 = (0, 1, 0)^T$$

  $$i_3 = (0, 0, 1)^T$$

  The identity operator I. is defined as a diagonal square matrix with unity elements on the diagonal and all other elements zero.

  Specific identity operators can be defined as were the specific identity vectors. For example, I_2. is a matrix of all zeros except for unity in the second diagonal position.

A.2 Coordinate System Definitions

A.2.1 General Conventions

As in the case with motion entities, rigorous definition of coordinate systems is required if an accurate dynamic analysis and subsequent computer implementation are to be accomplished.

For motion analysis, the standards used herein for coordinate systems incorporate:

* Right-handed orthogonal systems

* Specific axes in a coördinate system numbered 1, 2 and 3.

For example, if the column vector v is expressed in three elements resolved to the axes of a particular coordinate system, then the elements will have notation V_1, V_2 and V_3 . Note the upper case notation, since the elements of a column vector are scalars.

A.2.2 Abbreviations for Coordinate Systems

Many coordinate systems are usually required in a dynamic analysis. Abbreviating their full names is very convenient, especially in the checklist definitions of the various entities used in the analyses. Abbreviations take such forms as RAX for "rotor axes", IAX for "inertial axes", etc.

A.3 A Shorthand Notation for Entity Definition Checklists

The previous sections have expressed the powerful need for careful and comprehensive definitions of motion entities in dynamic analyses. Most motion entities require four items in the specification list: type, subject, base and resolution. Derivative entities require a fifth specification, the axis system in which the derivative is taken: the differentiation base.

Force and moment vector columns need type, subject (point of application) and resolution specifications.

A shorthand notation can be defined which simplifies laborious definitions for all items in an analysis. The shorthand also is a tool for discipline--it can be incorporated in each entity definition as a standard procedure, so that its presence assures proper and complete definition of each item used in an analysis. Consider the shorthand nomenclature

T :__, S :__, B :__, R :__ , D :__ .

abbreviations of various items are substituted where the "underline" areas appear in the suggested nomenclature. With the exception of the "type" part (T:) of the shorthand specification, the abbreviations usually refer to coordinate systems.

An example specification is

T:XV, S:CAX, B:IAX, R:CAX

Such a specification may apply to the vector, say v_c . In this case, v_c is a translational velocity of center-of-gravity axes (CAX) moving with respect to inertial axes (IAX) resolved to CAX. If a differentiation is involved, a fifth item is added. For example, $\dot{v}_r$ would be specified:

T:XV, S:CAX, B:IAX, R:CAX, D:CAX

In this case, the dot denotes differentiation with respect to time taken in the CAX system. Note that the type specification refers to the undifferentiated vector, in this case v_r . The dot is viewed as an operation on the entity, but the type of entity does not change because of the indicated presence of the operations.

To further aid in the shorthand process, the following abbreviations of entity types is suggested.

XP	Translational or linear position - a position vector
RP	Rotational position: i.e., an angular vector
XV	Translational velocity
RV	Rotational velocity
XA	Translational acceleration
RA	Rotational acceleration
F	Force
M	Moment

A.4 Operations on Scalars, Vectors and Operators

Standard texts on tensor mechanics define the types of operations that can be performed with scalars, vectors and tensors. Some of these operations are summarized below for completeness of this document, as they apply to the matrix representation of scalars, vectors and tensors.

A.4.1 Operations on Scalars

Straightforward arithmetic operations include addition, subtraction, multiplication, division; standard operations of the calculus include differentiation and integration.

A.4.2 Operations on Vectors

Column vectors can be added and subtracted by performing scalar additions or subtractions of their elements. If a and b are vectors

$$\begin{aligned} a + b &= (A_1 , A_2 , A_3)^T + (B_1 , B_2 , B_3)^T \qquad \text{(A4.1)} \\ &= (A_1 + B_1 , A_2 + B_2 , A_3 + B_3)^T \end{aligned}$$

The process of subtraction is identical.

Two types of vector products are defined, one which results in a scalar and one which results in another vector.

The scalar product is the product of the magnitude of two vectors times the cosine of the angle between vectors. For column vectors, the scalar product is the sum of the products of the elements. For the example vectors a and b previously defined, the scalar product is

$$S_{ab} = A_1 B_1 + A_2 B_2 + A_3 B_3 \ . \qquad (A4.2)$$

another form of this expression is

$$S_{ab} = a^T b \qquad (A4.3)$$

where a^T is a row matrix formed by transposing the a column matrix.

$$S_{ab} = S_{ba} \qquad (A4.4)$$

The vector product or "cross product" of two vectors is defined as a vector normal to the plane created by the two vectors being multiplied, with a magnitude equal to the product of the individual magnitudes times the sine of the angle between the operand vectors. If the two operands, say a and b, are column vectors, the cross product is conveniently expressed using a special cross operator,

$$V_{ab} = \text{cross product vector} = X(a).b$$

where the skew-symmetric matrix operator is defined

$$X\ (a).\ = \begin{bmatrix} 0 & -A_3 & A_2 \\ A_3 & 0 & A_1 \\ -A_2 & A_1 & 0 \end{bmatrix} \qquad (A4.5)$$

One important property of the cross product operation is expressed as

$$X\ (a).\ b = -X\ (b).\ a = X\ (-b).\ a \qquad (A4.6)$$

Another property is

$$X\ (a).\ a = 0 \text{ for all a.} \qquad (A4.7)$$

The argument indicated for X. in the definitions presented above is a vector. The argument can also be a scalar if X. carries a subscript indicating which positions the scalar is to occupy. Note the equivalence of the following situations in this regard:

X_i (A). is equivalent to X (a). with $a = Ai_i$

Other useful relationships associated with the cross product operator involve triple products:

$$X\ (a).\ X\ (b).c = (a^T\ c)b - (a^T\ b)c = S_{ac}\ b - S_{ab}c \qquad (A4.8)$$

$$X\ (a).\ X\ (b).\ X\ (a).\ = -a^T\ b\ X\ (a).\ = -S_{ab}\ X\ (a). \qquad (A4.9)$$

These relationships can be proven by processing the matrices and comparing results--element be element.

Division of vectors is generally not defined.

A.4.3 Operations on Second Order Tensors

Tensors are added and subtracted by element, as are vectors. Two operand tensors so processed must, of course, have the same dimensions (same numbers of rows and columns); the tensors need not be square.

Multiplication of tensors can occur in the form

$$A. = B.C. \qquad (A4.10)$$

but B. must have the same numbers of columns as C. has rows. If B. has i rows and j columns, and C. has j rows and k columns, then A. is of dimension i by k.

In general, the tensor product is not commutative, so B.C. is generally not equal to C.B.

A tensor can premultiply a vector to produce another vector, as in

$$a = G.b \qquad (A4.11)$$

This form is the standard form used in transformation procedures, where a vector resolved to one coordinate system is converted to a column representing the same vector resolved to another system.

Division of a tensor can occur if it is square and nonsingular. The process is called "inversion" in the matrix vernacular, and is denoted herein with superscript negative one (-1). If the inverse of a matrix, say M., exists, and

$$a = M.b \qquad (A4.12)$$

then

$$b = M.^{-1}a \qquad (A4.13)$$

The rules and methods associated with inversion are many: expression of these is beyond the scope of this document.

One rule on tensor products worth noting here is

$$(A.B.)^T = B.^T\ A.^T \qquad (A4.14)$$

where superscript "T" denotes the transpose.

A.4.4 The Calculus of Scalars, Vectors and Tensors

Scalars, vectors and tensors are called matrices, when expressed as arrays of scalar elements. All matrices can be differentiated and integrated by element. For example, the derivative of a column vector with respect to some parameter, say S, is vector whose elements are the derivative of the original vector elements with respect to S.

A.5 Standard Rotational Coordinate Transformations

Transforming motion and loads entities form one coordinate system to another is one of the most common procedures performed in solving dynamic system equations. Standardizing these processes, and using uniform notation, reduces the probability of error to a considerable degree. Also, standardization leads to relatively "clean" notation, simplifying the tasks of technical documention and computer coding.

In the author's experience, the poor and nonuniform selection of notation is a primary cause of error, second only to poor definition of motions entities and axes. Poorly designed notational systems lead to excessive use of multiple subscripts, superscripts, primes, hats, bars, etc., which lead to typographical and computer coding errors.

The purpose of this subsection are to define standard coordinate-transformation methods and to specify uniform nomenclature conventions for such processes.

A.5.1 Resolution Transformations

Since vectors do not require resolution, their definitions are not affected by coordinate frames not being parallel. Column-vectors, however, do require resolution for their definitions, and therefore require transformation.

Suppose, for example, that r_a is some position column vector, resolved to the a axis system (AAX), and r_b is a column vector representing the same entity as r_a but resolved to the b axis system (BAX). Then

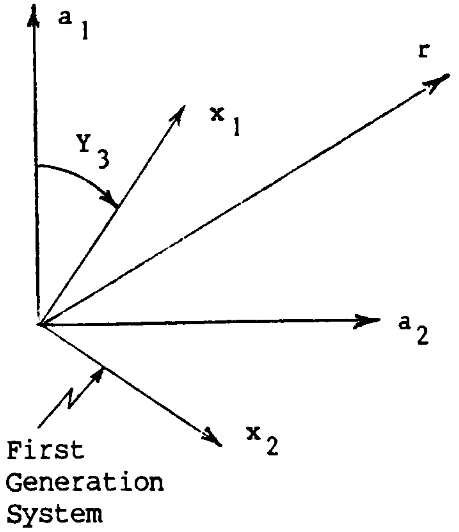
$$r_b = R.r_a \qquad \text{(A5.1)}$$

where R. is a 3 x 3 rotational transformation matrix. R. will be the identity matrix if the a and b axis systems are parallel.

The R. array is also called the matrix of direction cosines.

Many methods are available for calculating the matrix, R. . One convenient method was developed by Euler, and involves synthesizing R. using the product of three arrays which are easy to derive and remember.

Suppose one starts with AAX, and rotates about this system's number 3 axis an angle, Y_3 . A new system emerges; call it system x. Figure A.1 shows the process.



Note: Number 3 axes of both systems lie normal to the page pointing away from reader.

Figure A.1 Number 3 Axis Eulerian Rotation

Direct inspection of Figure A-1 reveals that, if R_{a1}, R_{a2} and R_{a3} are elements of the column vector r_a, resolved to AAX, then

$$R_{x1} = R_{a1} \cos Y_3 + R_{a2} \sin Y_3$$

$$R_{x2} = -R_{a1} \sin Y_3 + R_{a2} \cos Y_3$$

$$R_{x3} = R_{a3}$$

or in matrix notation

$$r_x = R_{a3} \ (Y_3). \ r_a \qquad \text{(A5.2)}$$

the number 3 rotation matrix is defined

$$R_3 \ (Y_3). = \begin{vmatrix} \cos Y_3 & \sin Y_3 & 0 \\ -\sin Y_3 & \cos Y_3 & 0 \\ 0 & 0 & 1 \end{vmatrix} \qquad \text{(A5.3)}$$

Subscript 3 was used on R. to denote rotation about the number 3 axis.

Now, without trying to envision the complete process in space, simply study the rotation from the x system to the second generation system, say y axes; the rotation is now about the number 2 axis of the x system and through an angle, Y_2. A figure similar to Figure A.1 can be prepared and the transformation equations again written by inspection. The result

$$r_y = R_2\,(Y_2).\,r_x \tag{A5.4}$$

where

$$R_2\,(Y_2). = \begin{bmatrix} \cos Y_2 & 0 & -\sin Y_2 \\ 0 & 1 & 0 \\ \sin Y_2 & 0 & \cos Y_2 \end{bmatrix} \tag{A5.5}$$

A similar process is followed for rotation about the number 1 axis of the y frame, through the angle Y_1 , to produce the third-generation system. if Y_3 , Y_2 and Y_1 have been selected properly, the third-generation system will be the b system. Hence

$$r_b = R_1\,(Y_1).\,r_y \tag{A5.6}$$

$$R_1\,(Y_1). = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos Y_1 & \sin Y_1 \\ 0 & -\sin Y_1 & \cos Y_1 \end{bmatrix} \tag{A5.7}$$

Combining Equations A5.2, 4 and 6, one sees that

$$r_b = R_1 \cdot R_2 \cdot R_3 \cdot r_a \tag{A5.8}$$

and by comparison with Expression A5.1

$$R. = R_1 \cdot R_2 \cdot R_3\,. \tag{A5.9}$$

(The arguments in the rotational arrays have been dropped in Expressions A5.8 and 9 for convenience.)
Equation A5.9 follows from Statement A5.8 since the elements in r are fully arbitrary.

It should be noted that, one axis system can be rotated to another by a series of separate rotations in any order. The 3-2-1 convention is standard in rigid body analysis such as used for aircraft simulation, but any number of rotations about axis numbers in any order can be used. Note that the orders and numbers of rotations only affect the definitions of the selected angles. The term "Euler angles", however, usually implies the order 3, then 2 and then 1, so that values specified for Euler angles carry this implied order.

This system for synthesizing R. is convenient because the individual (single-axis) rotations are easy to derive, easy to memorize, and convenient to synthesize from design drawings presented as orthographic projections. An important property of the orthogonal rotation matrix is

$$R_i\,(Y). = R_i^T\,(-Y). \tag{A5.10}$$

where i can be 1, 2 or 3. This property can be observed by inspection of Definitions A5.3, 5 and 7. Also, by direct calculation, it is easy to show that

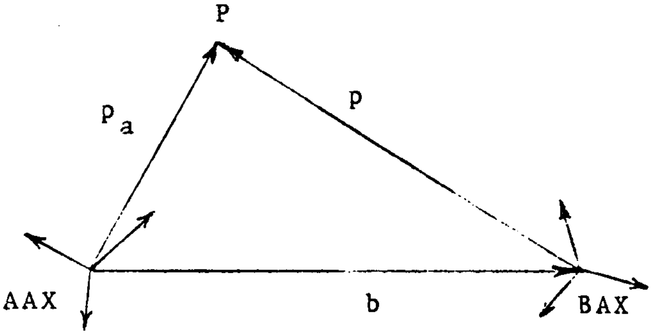
$$R_{i.}^{-1} = R_{i.}^{T} \qquad \text{(A5.11)}$$

A.6 Translational Transformations and Relative Motions

A.6.1 Motion Transformations

Relative transformations arise when motion and load entities are viewed from different coordinate frames (axis systems). For example, Figure A.2 shows the a and b axes (AAX, BAX), and a point P located positionally with respect to AAX by the vector p_a . P is located with respect to BAX by p and BAX are located with respect to AAX by b. If all three of these position vectors are resolved to the same axis frame, then

$$p_a = b + p \qquad \text{(A6.1)}$$



Definitions

p_a	- T:XP/S:P/B:AAX/R:BAX
p	- T:XP/S:P/B:BAX/R:BAX
b	- T:XP/S:BAX/B:AAX/R:BAX
v_p	- T:XV/S:P/B:AAX/R:BAX
v_b	- T:XV/S:BAX/B:AAX/R:BAX
w_b	- T:RV/S:BAX/B:AAX/R:BAX

Figure A.2 Relative Locations of a Point, P.

The translational velocity of P with respect to AAX is calculated by simply time-differentiating Equation A6.1.

$$v_p = d\ (p_a)\ /dT = d\ (b)\ /dT + d\ (p)\ /dT \qquad \text{(A6.2)}$$

where the derivatives are taken in AAX. If $p^{\cdot}$ is the derivative of p taken with respect to time in BAX, then a major theorem in vector analysis states that

$$d\ (p)\ /dT = p^{\cdot} + X\ (w_b).\ p \qquad \text{(A6.3)}$$

where w_b is the rate of angular rotation of BAX with respect to AAX, resolved to BAX.

A special condition arises when P is fixed with respect to BAX; for instance if BAX are fixed to a rigid body and P fixed to the same rigid body. In this case $p^{\cdot}$ is zero and Equation A6.2 and 3 can be combined to yield

$$v_p = v_b + X\ (w_b).\ p \qquad \text{(A6.4)}$$

where v_b is a direct substitution in notation for d (b) /dT. Equation A6.4 can be recast using Formula A4.6

$$v_p = v_b + X\ (-p).w_b \qquad \text{(A6.5)}$$

where now the cross product matrix X. is a constant because p is constant with respect to time (when viewed in BAX).

Equation A6.5 is essentially a translational transformation expressing the velocity of a point P, which has been translated from BAX the constant p, by transforming known BAX motion entities v_b and w_b.

The second derivative of p_a with respect to time taken in AAX is also of particular interest. If AAX is a Newtonion frame (i.e., AAX are fixed to the stars), then this second derivative becomes the acceleration of P. The second derivative is achieved by differentiating Equation A6.2 in AAX:

$$\begin{aligned} d\ (v_p)/dT &= d\ (v_b)\ /dT + d\ (p^{\cdot})\ /dT \\ &\quad + X\ (d\ (w_b)\ /dT).\ p + X\ (w_b).\ d\ (p)\ /dT \qquad \text{(A6.6)} \end{aligned}$$

Equations A6.3 can be applied to items in Equations A6.6 that involve differentials.

$$d\ (p^{\cdot}\ )\ /dT = p^{\cdot\cdot} + X\ (w_b).\ p^{\cdot}$$

$$d\ (w_b)\ /dT = w_b^{\cdot} + X\ (w_b).\ w_b$$

Noting that X (w_b). w_b is identically zero, the individual differential equations can be used to rewrite Equation A6.6 in the form:

$$\begin{aligned} d\ (v_p)\ /dT &= d\ (v_b)\ /dT \\ &\quad + p^{\cdot\cdot} + 2\ X\ (w_b).\ p^{\cdot} + X\ (w_b^{\cdot}).\ p + X^2(w_b).p \qquad \text{(A6.7)} \end{aligned}$$

Equation A6.7 is called the Coriolis theorem and is used extensively in Newtonian dynamic analysis.

Equation A6.5 can be differentiated in BAX to yield another transformation. If P is fixed in BAX, this result becomes

$$v_p^{\cdot} = v_b^{\cdot} + X\ (-p).\ w_b^{\cdot} \qquad \text{(A6.8)}$$

Equations A6.5 and A6.8 transform motion entities associated with BAX (moving with respect to AAX) into entities associated with P (moving with respect to AAX). Note that these transformations yield the translational motion entities for P. Rotational transformations are trival, since

$$w_p = w_b \qquad \text{(A6.9)}$$

$$w_p^{\cdot} = w_b^{\cdot} = d\ (w_b)\ /dT \qquad \text{(A6.10)}$$

In summary, key motion entities expressed for BAX moving with respect to AAX can be transformed to the same types of motion entities for point P which has been translated for BAX by the constant vector p. The necessary transformation equations are Expressions A6.5, 8, 9 and 10 which are linear because p is constant.

A.6.2 Loads Transformations

Consider loads entities defined as follows:

f - T:F/S:P/R:BAX
m - T:M/S:P/R:BAX
f_b - T:F/S:BAX/R:BAX
m_b - T:M/S:BAX/R:BAX

The column vectors f and m represent force and moment loads applied to P. The theory of statics shows that these apply equivalent loads acting at BAX given by the expressions

$$f_b = f \qquad \text{(A6.11)}$$

$$m_b = m + X(p).f \qquad \text{(A6.12)}$$

Equations A6.11 and 12 essentially transform f and m to act on a translated system, BAX. These transformations are analogous to motion entity transformations given by Formulae A6.5 and A6.8.

Rotational transformations used to change the resolution of a column vector work the same on loads columns, of course, as they do on motion entity columns.

A.7 Derivation of the Rigid Body Equations of Motion

The relationships presented in previous sections can be used to derive the general rigid-body equations of motion. Consider BAX of Figure A.2 fixed to the body. Let dM be an infinitesmal portion of the body mass loacated at point P.

The Coriolis theorem (Equation A6.7) states the acceleration of the mass under force df_b, so that Newton's second law of motion is expressed

$$df_b = dM\ d(v_p)\ /dT \qquad \text{(A7.1)}$$

If the force df_b is integrated throughout the body, the total force is calculated

$$f_b = M\ d(v_b)\ /dT + (X(w_b{}^{\cdot}). + X^2(w_b).)p_m \qquad \text{(A7.2)}$$

where constant zeroth and first moments of mass are defined

$$M = \int_V dM \qquad \text{(A7.3)}$$

$$p_m = \int_V p dM \qquad \text{(A7.4)}$$

The integrations are taken throughout the entire volume (V) of the body. If Bax are located at the body's center of gravity, then p_m vanishes.

Equation A6.12 shows that the moment at BAX due to force df on dM is

$$dm_b = X(p).df$$

Using the Coriolis theorem again with Newton's second law, and integrating throughout the body.

$$m_b = X(p_m).d(v_b)/dT + I_m.w_b^{\cdot} + \int_V X^2(w_b).pdM \qquad (A7.5)$$

where the second moment of mass or the inertia tensor is defined

$$I_m. = -\int_V X^2(p).dM \qquad (A7.6)$$

The integral term in Equation A7.5 can be rearranged to more convenient form, so that the final rotational equation of motion is expressed

$$m_b = X(p_m).d(v_b)/dT + I_m.w_b^{\cdot} + X(w_b).I_m.w_b \qquad (A7.7)$$

The rearrangement of the integral term is justified by applying the triple product relationship A4.8 to both forms used above, which will reveal the equivalance of the two formulations.

The term $I_m.w_b$ is the body angular momentum

$$h = I_m.w_b \qquad (A7.8)$$

So that the moment applied at the body center of gravity is the time derivative of the angular momentum (derivative taken with respect to inertial axes). Thus

$$dh/dT = h^{\cdot} + X(w_b).h \qquad (A7.9)$$

by application of Formula A6.3, so that Equation A7.7 becomes

$$m_b = X(p_m).d(v_b)/dT + dh/dT \qquad (A7.10)$$

The center of gravity of the body is located with respect to BAX by the vector c_g, so

$$c_g = p_m/M \qquad (A7.11)$$

With this definition the force and moment Equations A7.2 and A7.7 become

$$f_b = M(d(v_b)/dT + (X(w_b^{\cdot}). + X^2(w_b).)c_g) \qquad (A7.12)$$

$$m_b + MX(c_g).d(v_b)/dT + I_m.w_b^{\cdot} + X(w_b).I_m.w_b \qquad (A7.13)$$

A.8 Derivatives of Euler Angles

Subsection A.5 presents the Euler method for synthesizing the matrix of direction cosines used to change the resolution of a vector. For many dynamic analyses it is necessary to relate the time derivatives of the Euler angles to the rotational velocity of a coordinate system.

For example, let a be the column vector formed of three Euler angles needed for the direction cosine matrix between AAX and BAX of Figure A.2. Place the number-1 rotation as A_1, the number 2 rotation as A_2 etc.

If BAX has the rotational velocity w_b with respect to AAX, this velocity will give rise to dynamic Euler angles, $a^{\cdot}$. A relationship can be derived between w_b and $a^{\cdot}$ by viewing each element of $a^{\cdot}$, on at a time, and then by resolving these independent rate components to BAX. If $W_y.$ is defined

$$w_b = W_y.a^{\cdot} \qquad \text{(A8.1)}$$

then the transformation is expressed

$$W_y. = I_1. + R_1.I_2. + R_1.R_2.I_3. \qquad \text{(A8.2)}$$

where individual Eulerian rotation matrices are used to resolve each Euler rate element to BAX. The partial identity arrays $I_i.$ are used for convenience in selecting the desired elements of a in Expression A8.2.

The array $W_y.$ can be inverted by expanding Equations A8.1 and 2 and solving for $a^{\cdot}$. This process reveals the inverse of $W_y.$ to be

$$W_y^{-1}. = \begin{vmatrix} 1 & \tan A_2 \ \sin A_1 & \tan A_2 \ \cos A_1 \\ 0 & \cos A_1 & -\sin A_1 \\ 0 & \sin A_1 \,/\, \cos A_2 & \cos A_1 \,/\, \cos A_2 \end{vmatrix} \qquad \text{(A8.3)}$$

which can be confirmed by multiplication with A8.2 to yield the identity matrix.

APPENDIX B

AEROELASTIC ROTOR ANALYSIS

Page

Figures

Tables

The purpose of the analysis presented in this appendix is to derive a mathematical model for an aeroelastic rotor system. The model is intended to be applicable to rotors with very rigid blades (such as propellers) or rotors which have limber blades (such as helicopter and wind turbine rotors) where blade elastic motion has a significant effect on system operation.

The math model derived herein is intended to be incorporated into larger analyses, so that it becomes a component of a total simulation model. The rotor math model is needed to calculate the loads applied to the rotor supports by the blades, given the motion of the supports and control variables input to the rotor models.

The aeroelastic rotor analysis has been derived using procedures originally presented in Reference 5. The analysis presented below is complete, however, and should not require references other than standard texts on mechanics.

The models are derived viewing the rotor blades as massless elastic beams to which are attached blade "elements". Each blade element has an infinitesmal radial dimension, and is treated as a rigid body shaped as an airfoil. The blade element generates loads which are applied to the massless beam portion of the model, to complete the derivation of the partial differential equation representing the aeroelastic properties of the blade.

The D'Alembert approach to Newtonian dynamics is taken, where mass times acceleration is negated to become an apparent force applied to the massless structure. The acceleration of the blade element is calculated given the motion entities for the gimbal ring and the flexible motions of the massless beam. This acceleration applied to the inertial properties of the element is given a negative sign to become the "inertial loading" of the blade.

The aerodynamic environment in which the blade element is operating (i.e., the airspeed, Mach number and angle of attack) is calculated in the analysis. It is presumed that some aerodynamic definition for the blade is defined, usually in tabular form, that will produce the nondimensional aerodynamic loading coefficients (lift, drag and moment coefficients) as a function of the aero environmental parameters. The coefficients are then further processed to yield dimensional loads applied to the element due to aerodynamics.

The inertial and aerodynamic loads are summed for all elements in the model to produce forcing functions which act on the blade flexibility characteristics, and which also sum to the final loads applied to the gimbal ring by the blades.

The blade motion problem is of particular importance in rotors with limber blades. The "modal" analysis method is used herein, where the basic blade model derived as a mass-elastic continuum is transformed to "normal coordinates" which are functions only of time. This approach minimizes the number of degrees of freedom that must be incorporated in an analysis to calculate blade motions within a prescribed accuracy. The method is to be distinguished from the "nodal" method used in some analyses, which requires many more degrees of freedom than the modal analysis to achieve the same accuracy.

Further sections of this derivation contrast the modal and nodal methods of structural analysis in more detail.

A primary difference between the material presented below and that of the references is nomenclature. The nomenclature is all new and has been prepared for ease in word processing and data base management systems. The nomenclature has also been defined so that it relates very closely to program instructions that are actually executed by computer.

Appendix A describes the nomenclature conventions used throughout the analysis. In addition, a number of analysis methods are defined in Appendix A that are used throughout the presentations below without further definition.

Table B.1 defines the nomenclature used in the derivations below.

B.1 Fundamental Approach- Coordinates

A major portion of the analysis simply involves definition of suitable coordinate systems followed by routine application of the methods documented in Appendix A. Table B.2 summarizes the coordinate systems used throughout the presentations.

The analysis begins with the definition of a "reference point" which lies on the rotor shaft centerline near the center of the rotor hub. A nonrotating coordinate system called "hub axes" (HAX) has its origin at this reference point. The motions of HAX with respect to the stars (called "inertial axes" herein- IAX) are taken as given entities to the rotor analysis. These motions are to be given as translational and rotational inertial velocities, and the time derivatives of these entities taken with respect to hub axes. Additionally, the translational and rotational velocities of HAX with respect to the air mass in the vicinity of the rotor are also given to the analysis.

The "rotor axis system" (RAX) is defined with its origin and number-3 axis coincident with the corresponding HAX entities. RAX spin with the rotor, however; they have a rotational rate of Z radians per second with respect to HAX- the spin vector being coincident with the number-3 axes of HAX and RAX. For consistency with past nomenclature conventions, a positive Z is associated with RAX moving with respect to HAX about the negative number-3 axis. If z is the rotational vector representing the rate RAX moves with respect to HAX, then z is defined

$$z = (0,0,-Z)^T \qquad \text{(B1.1)}$$

The spin acceleration nomenclature used herein is $Z^{\cdot}$.

The analysis assumes that a gimbal system is installed between the shaft and an outer gimbal ring. The blades in the rotor are rigidly mounted to this outer gimbal ring. The gimbal analysis defines the motion of this outer gimbal ring with respect to hub axes, to that outputs from the gimbal analysis are inputs to the rotor blade analysis. The gimbal system may be of any configuration such as teetering (see-saw rotors) or floating (e.g. a ball joint).

Appendix C presents the gimbal analysis. The gimbal equations transfer the motion entities of hub axes to "shank axes" SAX, and then transfer forces and moments applied to the gimbal ring by the blades back to hub axes. Viewed in this way, the gimbal analysis is an interface between the rotor (including the outer gimbal ring) and the rotor support system which defines the motions of HAX.

The shank axis system for a blade is fixed to the outer gimbal ring. If the gimbal is undeflected, the number-3 axes of hub and shank axes are considered coincident, as are the origins. A separate shank axis system is defined for each blade in the rotor, so that each blade shank lies at the origin of its associated SAX system. The projection of the blade on the shaft normal plane lies generally along the negative number-1 SAX axis, to conform to conventions of previous rotor analyses.

Since the analysis of Appendix C has treated the transformations through the gimbal interface, the task of this present document involves the analysis of the blades mounted to the gimbal ring. The objective is to calculate the forces and moments (i.e., the loads) applied to the support system located at HAX by the rotor system, given the HAX motion entities, the rotor spin rate and acceleration, and the controls applied to the system.

B.2 Blade Coordinates

Aeroelastic rotors are so called because the blades deflect enough to change the loads to a considerable degree. Accordingly, it is necessary to calculate the motions of the blades before the loads can be computed. These motions must be defined in terms of suitable coordinates.

Figure B.1 shows the major coordinate systems incorporated in the aeroelastic blade analysis. The shank axes are depicted toward the hub end of the system. The "blade reference line" (BRL) moves along the span of the blade, being defined by the locus of all points "P", one of which is shown in the Figure.

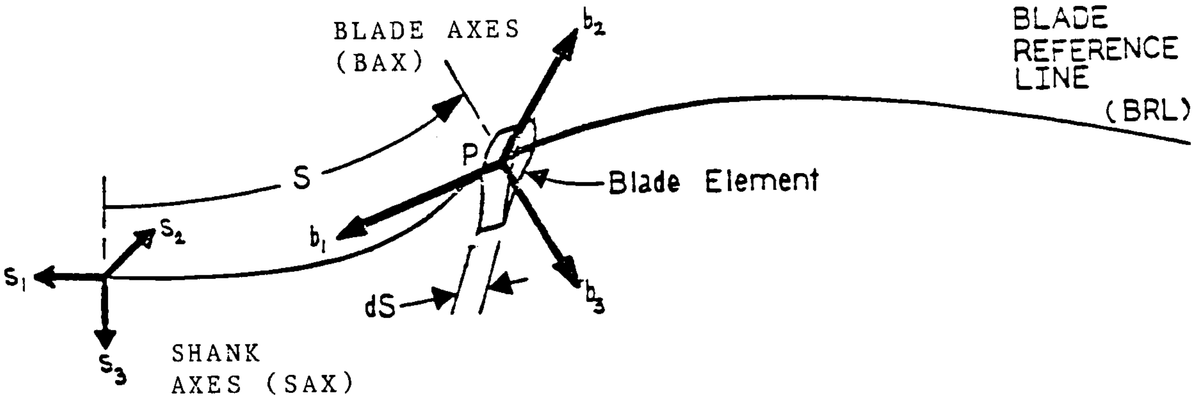


Figure B.1 Blade Reference Line Geometry

The BRL can be located within each section of the blade at any arbitrary point. The quarter-chord point is often chosen because this is the aerodynamic center for conventional airfoils. Other points (for example, the locus of blade-section centers-of-gravity) could also be chosen.

The BRL is defined as made up of specific mass molecules located in the blade. This abstraction allows the definition of a line coordinate, S, which will always be bounded between 0 and the total length or radius of the blade, say R. When the blade is under loading it can bend in many directions. Regardless, a specific value of S will locate a specific point, P, which is attached to a specific molecule in the blade at all times. The actual number of S is defined to be the line integral distance between the shank axis system origin and the point P, when the blade is fully unloaded. After any loads are applied, the line integral between SAX and P will, of course change, but the value of S identifying P will remain the same.

The parameter S is used throughout the analysis as the variable of integration; many functions that vary with blade radial position (and often time) are integrated as S moves from 0 to R. The BRL coordinate concept outlined above precludes S itself ever being a function of time.

The "blade axes" (BAX) are located with origin at P, with the number-1 axis tangent to the BRL at P and pointing generally toward SAX. Like the BRL, BAX are fixed to the mass molecules at P, so the locations of these coordinates are always defined. The specific angular orientation of BAX (about the number-1 axis) can be specified in any convenient manner. Locating BAX on each blade section so the number-2 axis is parallel to the blade chordline is one convenient definition, but others are possible. Some analysts prefer orienting BAX with respect to the airfoil line of zero lift, for example.

With the definitions of the BRL and BAX complete, the blade motion problem comes to defining the motions of BAX at every S as a function of time. The next subsection discusses the fundamental approach taken toward this portion of the analysis.

B.3 BRL Shape Definition

The "blade motion" problem is needed to define the shape of the BRL, since BRL motion often has a significant effect on loads applied to the supports by the complete rotor system. This subsection defines methods and nomenclature used to define the geometry of the BRL.

The vector, b, locates the point P (at station S) with respect to SAX. The resolution of b is also SAX. This position vector is, of course, a function of S,T where T is time.

The vector, f, is used to refer to a column of control parameters that mechanically change the blade structure. On most rotors only a single parameter "feathers" the blade; hence the choice of "f" to represent the "mechanical configuration" column. A column is used in the analysis for generality, because multiple mechanical controls have been implemented in some past rotor designs.

The array R. is used to convert vectors resolved in SAX for resolution to BAX; this is the matrix of direction cosines defined in Appendix A.

This analysis incorporates concepts of "quiescent" or "initial" conditions of the BRL to support the validity of certain assumptions to be made later in the derivations. The rotor blade is first viewed as spinning at some nominal rate under a prescribed airload. These nominal conditions are usually chosen to lie somewhere close to the middle of the blade operational profile, so that BRL motions from a quiescent shape associated with these nominal conditions are small. When operating in the defined nominal or quiescent conditions the geometry of the BRL is defined by the "initial shape" versions of b and R., denoted b_{oo} and R_{oo}. herein. Under the quiescent conditions, the rotor spin rate is Z_o, the mechanical control is f_o and the spin acceleration is defined as zero.

Now consider a change of f from f_o while the blade is otherwise in its quiescent condition. The BRL will move to a new position defined by the nomenclature b_o and R_o.. A transformation array R_f. can be defined which rotates the BAX from the nominal (initial shape) position to the intermediate initial shape position. The direction cosine arrays are related as follows-

$$R_o. = R_f.R_{oo}. \tag{B3.1}$$

The shape of the BRL after movement of f from the initial value will be called the "controlled shape" of the blade. Note that the initial shape is a constant in time but the controlled shape varies with time if the column f does.

Elastic motions due to applied loading other than the defined quiescent loading will move to BRL to its final position at any instant of time. The column vector b refers to the final absolute position of the BRL at any time, so that the positional elastic deformation is $b-b_o$. An "elastic angular" deflection column, e_a, is also defined. This column contains three Euler angles which define the matrix of direction cosines R_e.. R_e. transforms a vector resolved to BAX in the controlled position to reference BAX in the final instantaneous position of the BRL. The rotations are defined-

$$R. = R_e.R_o. \tag{B3.2}$$

It is convenient for some portions of the analysis to group all six elastic deformations into a single vector, w_e; in the conventions of Appendix A-

$$w_e = ((b-b_o)^T,e_a{}^T)^T \tag{B3.3}$$

The column vector w_e completely defines the shape of the BRL given the controlled position of this coordinate. w_e is the deflection entity which arises because of the application of loads to the BRL, other than the loads defined for the quiescent condition.

Note the resolution of the elements in w_e. The first three elements are resolved to SAX, and the last three are resolved to BAX in the controlled position. These resolution conventions are important, and will be used when the model for BRL elasticity is defined.

B.4 Loads

As stated previously, loads models are derived using "blade element" or "strip" theory. In this approach the blade is viewed as a massless elastic body to which are installed an infinite number of blade elements. Each blade element has mass and is usually shaped like an airfoil (except near the root for most rotors) so that aerodynamic loads are applied to the element of infinitesmal radial dimension.

The blade element supports applied loads from inertial and aerodynamic origins, and applies these to the massless BRL. The element has radial dimension, dS. A distributed force, f, and moment, m, can be envisioned, so that the force and moment vectors supported by the element are fdS and mdS, respectively. Subscripts "i" and "a" are applied to f and m to denote "inertial" and "aerodynamic" origins, respectively.

For consistency with the BRL geometric entities defined above, distributed force vectors are resolved to SAX and distributed moment columns are resolved to the control-positions of the BAX.

As was the case with the BRL geometric coordinates, it is convenient to place the force and moment distributed loading columns into a single loading column, p -

$$p = (f^T, m^T)^T \tag{B4.1}$$

The symbol p_o denotes the p column applied to the BRL for the defined quiescent loading conditions. This quiescent loading is made up of an aerodynamic contribution and an inertial distributed load due to centripetal acceleration.

The acceleration of the blade element is calculated based on the motion of the BRL (produced by the blade-motion portion of the analysis) and on the motion of SAX produced by the gimbal analysis. The method of D'Alembert changes the sign of this acceleration and views the result (multiplied by the mass of the element) as an applied force. This force must be supported by the massless BRL to produce the calculated acceleration. Viewed as such, this force is the "inertial load" on the massless BRL.

The mass of blade element is defined as $M_b dS$, where M_b is the distributed mass of the BRL, a function of S. The blade element also has a distributed inertia, $I_b..$, so that the rigid body rotational inertia of the blade element is $I_b..dS$. Most of the elements in this distributed tensor vanish because of the vanishingly small radial dimension of the element, dS. Only elements that do not contain products of the number-1 BAX dimension (i.e., the dS dimension) are nonzero in $I_b..$.

The other primary source of loading on the BRL is aerodynamic. The motions of SAX and the BRL cause the element to move with respect to the air mass in the vicinity of the element, and this motion gives rise to aerodynamic distributed loads on the blade element. The aerodynamic equations can be written for the airfoil of the blade element (with infinitesmal span, dS). Appropriate aerodynamic math models can then be used to define the distributed aerodynamic loading functions given the relative motions of the element. The details of typical aero models are described in a later subsection, below.

The distributed loads on the BRL cause elastic deformation which is the blade motion problem. The loads also sum to produce the total force and moment vectors applied by the BRL to SAX. In the analysis, these columns are given the symbols f_s and m_s, respectively.

B.5 The Elastic Model

Rotor blades can have very complex structural designs that give rise to irregular structural properties. Many rotors have one or more hinges installed for control and stress relief at certain radial stations: such hinges introduce singularities in the stiffness properties of the blade about the axes of the hinges.

The mathematics of structural analysis is presented in many standard texts. A conventional vehicle for definition of the elastic properties of a structure is the "stiffness influence function" or "kernal function", K_e.. In BRL geometric notation defined above, the stiffness model for the BRL can be expressed-

$$p(S) - p_o(S) = \int_o^R K_e(w_e, f, S, N) . w_e(N) \, dN \qquad (B5.1)$$

where N is a dummy variable of integration substituted for S. Note that the elastic deflection coordinate, w_e, is incorporated as an argument in the influence function. This indicates that the structural properties of most whirling rotor blades are very nonlinear. The nonlinearity precludes the usual and convenient assumption that deflections do not affect loads. The loading, particularly due to the centrifugal forces supported radially by the whirling blade, contribute significantly to the stiffness properties. As the rotor spin rate Z changes, the BRL shape changes which reflects the change in centrifugal loads supported by the structure. These loading changes modify the effective stiffness of the structure.

Also note the presence of the feathering control column, f, in the stiffness function argument. Mechanical control positions portions of the blade with respect to other portions; such geometrical changes will produce significant changes in the stiffness properties.

Equation B5.1 shows the loads and displacement columns as functions only of S. In a dynamical situation, p, w_e and f all become functions of time as well as the S radial coordinate.

This analysis assumes that appropriate definition of the stiffness influence function is provided as data defining the blade properties. The need for this definition is minimized in later sections of this document, however, as the parameters associated with eigenanalysis are effectively substituted for the most significant portions of the structural influence. Refer to Subsection B.8 for further discussion of these substitutions.

B.6 Inertial Distributed Loads

As stated previously, the distributed inertial loads are actually accelerations (times distributed mass) viewed as applied loads in the method of D'Alembert.

The blade element at station S is viewed as a rigid body. The equations of rigid-body motion have been derived in Appendix A, Subsection A.5.4. These equations can be applied directly to the blade element situation to compute the force and moment on the element. Changing the signs on these distributed force and moment vectors produces the loads applied to the BRL.

The following expressions arise from these applications of the derivations of Appendix A (Refer to Table B.1 for nomenclature definitions)-

$$f_i = -M_b \; (u_{br} + R^T . \; (X(w_b^{\bullet}) . + X^2 \; (w_b) . \;) \; c_g \;) \qquad \text{(B6.1)}$$

$$m_i = -M_b \; X(c_g) . \; R . \; u_{br} + I_b . \; w_b^{\bullet} + X(w_b) . \; I_b . \; w_b \qquad \text{(B6.2)}$$

$$u_{br} = d(v_b)/dT$$

or, restated in expanded form-

$$u_{br} = u_s + b^{\bullet\bullet} + 2X(w_s).b^{\bullet} + X(w_s^{\bullet}).b + X^2(w_s).b \qquad \text{(B6.3)}$$

$$u_s = v_s^{\bullet} + X(w_s) . \; v_s \qquad \text{(B6.4)}$$

$$p_i = (f_i^T, \; m_i^T)^T \qquad \text{(B6.5)}$$

B.7 Aerodynamic Distributed Loads

The methods of Appendix A can be applied to compute the velocity of a rotor blade element with respect to the air in the vicinity of the element. The result (in the nomenclature of Table B.1) is-

$$v_{ar} = R . \; (\; v_{as} + b^{\bullet} + X(w_{as}) . \; b \;) \qquad \text{(B7.1)}$$

Note that this vector is resolved to BAX.

This calculation of the aerodynamic velocity vector of BAX views the air mass surrounding the rotor as moving as a rigid body: this air body can have three translational and three rotational velocity components of motion with respect to inertial space.

Mathematical models associated with aerodynamic interference and wind must be used to compute the air mass motion- these models are viewed as outside the rotor models in this derivation. (Aerodynamic interference velocities are air motions created by devices submerged in the air mass that support aerodynamic loads).

With some analyses this rather simplified view of air motion is not adequate, and additional modes of motion must be allowed for the air mass. In these cases suitable models must be derived and used to compute the air motions. The results of such computations are often added directly to v_{ar} at each radial station for each blade in the rotor. The expression for the total aerodynamic velocity is written-

$$v_a = v_{ar} + v_{aa}$$

where v_{aa} is the "augmentive" local aerodynamic velocity added to the basic calculation to produce the final result. As stated above, this augmentive interference component must be supplied to the rotor analysis from outside models dealing with the complexities of aerodynamic interference velocities.

The Mach number and angle-of-attack of the blade airfoil at S are computed directly from elements of v_a-

$$M^2 = (V_{a2}^{\ 2} + V_{a3}^{\ 2}) \ / \ V_c^2 \qquad (B7.3)$$

$$A_b = \tan^{-1} (V_{a3}/V_{a2}) \qquad (B7.4)$$

where V_c is the velocity of sound. The aerodynamic pressure is given by the expression-

$$Q_a = 1481 \ M^2 \ P_r \qquad (B7.5)$$

where P_r is the ambient pressure ratio: the local atmospheric air pressure divided by sea-level standard pressure (2116 psf). Classical aerodynamic theory shows that the aerodynamic forces produced by a body in an airstream are proportional to this aerodynamic pressure times a characteristic area. Therefore, the force on an infinitesmal blade element divided by the element radial dimension (i.e., the force per unit radius or distributed force) is proportional to the dynamic pressure times the chord. A convenient constant of proportionality for the loads expressions is thus stated-

$$Q_{ac} = Q_a C_b \qquad (B7.6)$$

The specific characteristics of the airfoil at a blade station will give rise to nondimensional coefficients of force and moment, which might be expressed as functional relationships as follows-

$$c_{fa} = c_{fa} (A_b, M) \qquad (B7.7)$$

$$c_{ma} = c_{ma} (A_b, M) \qquad (B7.8)$$

Note that the functions indicate dependence on angle-of-attack and Mach number. Some airfoils may have other dependencies. For example, if a slot is incorporated in the blade from which an air sheet effluxes, then the momentum coefficient of this efflux will also influence the force and moment loads coefficients.

Specific models for the loads coefficients will have to be prescribed for an analysis before the rotor characteristics are defined. These prescriptions often take the forms of data tables that can be interpolated to yield the necessary functional definitions.

Assuming that a suitable numerical process yields the necessary aerodynamic coefficients, the distributed aerodynamic force and moment columns for the blade element can be defined-

$$f_a = Q_{ac} \, R_{\cdot}^{T} \, R_{1}^{T} \, (A_{bl}) \cdot c_{fa} \qquad \text{(B7.9)}$$

$$m_a = Q_{ac} \, C_b \, c_{ma} \qquad \text{(B7.10)}$$

and the final distributed load column stated-

$$p_a = (f_a^T, \, m_a^T)^T \qquad \text{(B7.11)}$$

These expressions when combined with a suitable tabular (or other) definition of the blade element aerodynamic properties suffice to define the aerodynamic distributed loads.

B.8 Blade Motion Analysis

B.8.1 Nonlinear Blade Motion Equation

Equation B5.1 is the elastic model for the massless BRL. The distributed load column, p, arises from inertial and aerodynamic phenomena. Adding these contributors and eliminating p produces the generalized blade motion equation-

$$p_i(S, T) + p_a(S, T) - p_o(S) = \int_o^R K_e(w_e, f, S, N). w_e(N, T)dN \qquad (B8.1)$$

The solution approach taken in this analysis is to transpose the loads columns to the right hand side of the equation, and then to add terms representing linear components of the inertial loading function and elastic property to both sides of the expression. The result is-

$$M. \ddot{w}_e + K_I. w_e + \int_o^R K_{ep}(S, N). w_e(N, T) \, dN = p_n(S, T) \qquad (B8.2)$$

$$p_n = p_i + M. \ddot{w}_e + K_I. w_e + p_a - p_o - \int_o^R (K_e. - K_{ep}.) \, w_e \, dN \qquad (B8.3)$$

where now, p_n is the "nonlinear" distributed loading function. The arrays M. and K_I. are linear portions of p_i associated with the relative acceleration (i.e., acceleration of the BRL with respect to SAX but not the total acceleration with respect to inertial axes) and an inertial stiffness. The inertial stiffness arises because the mass of the blade moves within a conservative centripetal acceleration field, creating the apparent inertial forces that appear as stiffness terms. Note that these linear mass and stiffness operators are both functions of S.

The "perturbation" stiffness influence function K_{ep}. is a linearized form of K_e. Note that the linearized stiffness function has the BRL elastic deformation coordinate and the feathering column removed from its argument.

No assumptions have been made in equations B8.2 and B8.3. The same items have simply been added to both sides of the equation. The items have been chosen to represent the most powerful mass and stiffness terms in the equation, however, so that the added terms on the right hand side of the equation effectively subtract the linear mass and stiffness properties from the full nonlinear representations of these entities. The result is a nonlinear forcing function on the right hand side of the equation with very weak (and in some cases, negligible) mass and stiffness influences. Of course the modified distributed loading column still has strong influences due to BRL rates, aerodynamic loads (which tend to relate mostly to BRL rates and not position or acceleration) and inputs to the rotor analysis due to HAX motions and control.

Many standard texts derive Maxwell's reciprocity theorem for structural influence functions. Since the perturbation function in the above equation is now linear (it does not depend on deflections or loads) this theorem applies and can be expressed-

$$K_{ep}(S,N). = K_{ep}^{T}(N,S). \qquad (B8.4)$$

This relationship arises because of the principle of energy conservation within a conservative (linear) structure; the principle is essential to major conclusions made in the normal mode analysis presented later.

The next subsection addresses analysis methods that are to be applied to the reformulated blade motion equation, Expression B8.2.

B.8.2 Nodal vs Modal Analysis Methods

Equation B8.2 is a partial integro-differential equation with two independent parameters, S and T. The equation is linear if the nonlinear forcing function, p_n is viewed as a driving function provided outside of the solution of the basic equation itself.

The approach taken herein is to solve the blade motion equation numerically. The results of the solution provide all entities required to solve for the forcing function, which is then applied to the equation for a short time duration. The equation is solved for new BRL motion entities, and the process is repeated in a loop.

The equation is still in the form of a continuum, and therefore represents an infinite number of degrees of freedom. This form must, of course, be converted to a finite number of coordinates for practical solution.

Two basic methods are often used so solve such equations, the finite-element or "nodal" approach and the eigenfunction or "modal" approach. As stated before, the second of these has been chosen for the present derivation, but the paragraphs below present the trades that were made in arriving at the chosen method.

The Nodal Method

In this approach, the equation is reformulated to apply to a finite number of rigid bodies; each pair of adjoining bodies is connected with a massless beam. The bodies are called nodes, and are each given up to six degrees of freedom. The equations for the nodal model can be derived by converting equation B8.2 into a summation equation instead of an integral expression, facilitated by allowing each blade element (node) to have a finite radial dimension.

The nodal model so derived can be expressed as a second order matrix equation in constant coefficients. The entire equation can be integrated numerically (solving for the motions of all of the nodes) to represent the flexible blade motion problem.

The major problem with this approach is the number of nodes required to get good accuracy. If the model is to be a high fidelity representation of the blade dynamics for, say, the frequency range associated with the first two natural modes of motion, up to 30 nodes will be required to represent the natural frequencies accurately. Such a model will have up to 180 degrees of freedom, just to reproduce two physical modes with reasonable accuracy.

When the equation is solved numerically, the numerical step size (on time) needed to maintain numerical stability of the solution will be governed by the highest mode in the model, in this case the 180 th mode. This mode is, of course, not of interest to the solution, but it will have to be treated properly or the numerical solution will go unstable even if the physical model is stable.

This situation causes the nodal approach to the forced-motion solution to be very expensive and often totally impractical. The nodal model can be used to calculate the natural frequencies and modeshapes for the blade, however, and these resulting "eigenfunctions" can then be used to perform a modal analysis for the forced-motion problem.

The Modal Analysis Method

In this method, the continuous dynamical representation of the blade structure is transformed into an infinite number of equations, each describing the participation of a single natural mode of vibration in the overall dynamical response of the blade. This infinite number of total differential equations in time can then be reduced to a small finite number of equations by rationalizing which of the natural modes will remain unexcited for the particular analysis being conducted. The analyst should know the frequency content of the forcing function on the blade (i.e., the frequencies of interest in the function p_n). Knowing this, a rational rejection of modes above certain natural frequencies can be made because it will be known that these higher modes will not participate in motions under the function p_n.

In this way, a minimum number of modal degrees of freedom can be used in the forced-motion numerical integration, with predictably accurate results. The approach will be a minimum cost approach considering the expense of data processing time when the model is executed in a computer.

With these background considerations in place, the next step in the analysis process is the eigensolution of Equation B8.2- the subject of the next subsection.

B.8.3 The Eigenproblem

The Eigenproblem deals first with the solution of the homogeneous part of Equation B8.2, i.e., the expression with $p_n = 0$. The equation is now linear with two independent variables, S and T.

The usual approach to solution of such equations is to propose a solution form which is the product of two functions, one a function only of S and the other a function only of T. In this case, a simple sinusoidal function is proposed for the portion dependent on time.

$$w_e = w_i \sin W_i T \qquad \text{(B8.5)}$$

This solution form can be substituted into the homogeneous equation to become:

$$-W_i M. w_i + K_I. w_i + \int_o^R K_{ep}(S, N). w_i(N)\, dN = 0 \qquad \text{(B8.6)}$$

This is called the characteristic equation which requires eigensolution.

There will be solutions to Equation B8.6, i=1,2,.. infinity, each with a specific value for the eigenvalue W^2_i and each with a specific function $w_i(S)$ called the eigenfunction or modeshape. These solutions are subject to the boundary conditions placed on the BRL, which in this case are zero position and slope at the root and zero shear force and moment at the tip.

Many numerical methods are available for calculating the eigenvalues and eigenfunctions for a structure such as this rotating beam system. The methods almost all involve reformulating Equation B8.6 into a nodal or finite element model and then processing the resulting matrix equations. Sophisticated programs such as NASTRAN have been developed over many years with facilities for preparing eigensolutions for complex structures.

For the present analysis it is assumed that the eigensolutions are available for use in formulating this forced-motion solution method. In a later subsection it will be seen that the eigenfunctions essentially replace the model for the structural stiffness influence function. The ramifications of this substitution will also be discussed in that section.

Before leaving the subject of the eigenproblem, an important characteristic of the eigenfunctions will be demonstrated: their orthogonality with respect to the structural mass. To show this, consider two representations of Equation B8.6, one for the i'th mode (i.e., with eigenparameters W^2_i and w_i incorporated) and the second with the j'th mode. Premultiply the first equation by the transpose of w_j and the second expression by the transpose of w_i, subtract the equations and integrate the result on S between 0 and R. Each term in this equation is a scalar, so any of the terms can be transposed without changing their value. Also, the mass and inertial stiffness matrices are known to be symmetric, so they equal their own transposes. The results of these manipulations produce the following equation:

$$-(W_i^2 - W_j^2) \int_o^R w_j^T M. \, w_i \, dS$$
$$+ \int_o^R \int_o^R [w_j^T (S) \, K_{ep}(S,N). \, w_i(N) - w_j^T(N) \, K_{ep}^T (S,N). \, w_i(S)] \, dN \, dS = 0 \qquad (B8.7)$$

Recalling Maxwell's reciprocity relationship B8.4, it is seen that the double integral terms incorporating the stiffness cancel: this occurs because the order of integration on S and N can be reversed. Since the eigenvalues are different for each mode (distinct) for most structural problems, the following relationship emerges:

$$\int_o^R w_j^T M. \, w_i \, dS = 0 \quad , i \neq j \qquad (B8.8)$$

This is the orthogonality expression on the system mass matrix.

The modeshapes are also orthogonal with respect to the system stiffness. This can be seen by premultiplying Equation B8.6 by w_j transpose and integrating over the entire blade radius. Since the mass term vanishes when i is unequal to j, the resulting stiffness integrals must also vanish for unequal i and j- effectively expressing the orthogonality of the eigenfunctions with respect to the system stiffness model.

$$\int_o^R [w_j^T K_I. \, w_i + \int_o^R w_j^T (S) \, K_{ep}(S,N). \, w_i(N) \, dN] \, dS = 0 \quad , i \neq j \qquad (B8.9)$$

These special characteristics of the eigenfunctions distinguish them from any other functions that may be chosen for the model, in that they enable decoupling the equations of motion for the system. This process will be demonstrated in the next subsection.

B.8.4 Coordinate Transformation Using the Eigenfunctions

Now propose a coordinate transformation for Equation B8.2 of the form-

$$w_e \,(S,\, T) = \sum_i w_i(S)\; Q_i(T) \qquad \text{(B8.10)}$$

This expression can be substituted into Equation B8.2, and the resulting formula premultiplied by the transpose of w_j and integrated over the radius. The result is-

$$\sum_i \int_o^R [w_j^T M .\; w_i \ddot{Q}_i + w_j^T K_I .\; w_i Q_i$$

$$+ \int_o^R w_j^T \,(S)\; K_{ep}(S,N) .\; w_i(N)\; Q_i \; dN]\; dS = \int_o^R w_j^T\, p_n \; dS \qquad \text{(B8.11)}$$

Because of the orthogonality characteristics on the mass and stiffness properties that were demonstrated in the previous subsection, all terms in this equation vanish except when i=j. Additionally, for those remaining terms, Equation B8.6 can be used to eliminate the integrals involving the stiffness function in lieu of terms involving the system mass and the eigenvalues. The resulting equation is

$$M_{gj}\; (\ddot{Q}_j + W_j^{\,2}\; Q_j) = G_{gj} \qquad ,\; j = 1,2,... \qquad \text{(B8.12)}$$

where the generalized mass and generalized forcing functions are defined by the following two expressions, respectively:

$$M_{gj} = \int_o^R w_j^T M .\; w_j \; dS \qquad \text{(B8.13)}$$

$$G_{gj} = \int_o^R w_j^T\, p_n \; dS \qquad \text{(B8.14)}$$

Note that the linear portion of the structural stiffness influence function has vanished completely from the model, although nonlinear stiffness effects still remain in the forcing function (Equation B8.3). This is the process mentioned previously, where the structural property is effectively replaced by the eigenfunctions.

Equations B8.12 are a rigorous representation of the original continuum because there are an infinite number of them. An infinite number of independent functions will represent any arbitrary shape of the BRL, so the solution at this point still invokes no assumptions. All nonlinearities have been retained in the nonlinear distributed loading function, p_n.

Of course it is impractical to solve for an infinite number of equations, so the basic assumption on the solution involves restricting the number of normal mode equations (Expressions B8.12) to only those of interest in a particular forced-motion solution.

The next subsection presents a number of important observations regarding the modal representations derived above.

B8.5 Observations on The Decoupled Model

As mentioned before, the decoupled normal mode equations allow the analyst to have considerable insight into the physics of the problem, in order to make good judgments on selecting which portions of the model to retain and which to discard. The modal equations reflect the natural modes of vibration of the system which will always characterize the response of the system under the influence of external loads.

To explore this in more detail, consider Figure B.2 which shows the dynamic response of one of the Equations B8.12 to the application of sinusoidal loading with varying frequency. This Figure is often considered a plot of the "dynamic amplification factor", revealing how the system responds to stimulation.

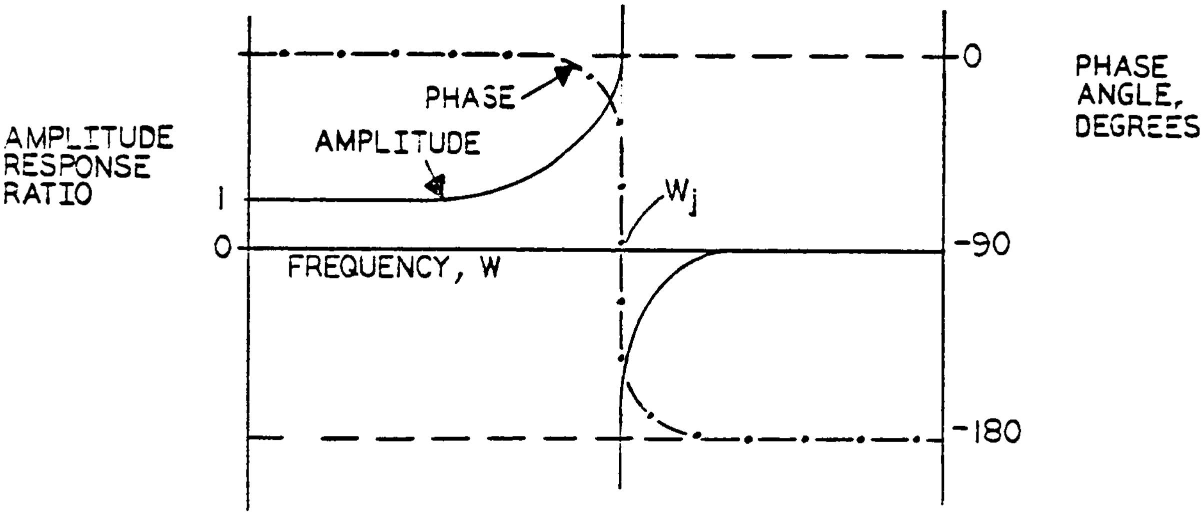


Figure B.2 Frequency Response Characteristics of the Modal Equations

The main interest here is when the resonance frequency, W_j is large compared to the frequency of excitation. When this is the case, the figure shows that the equation responds with an amplitude near that exhibited with forcing functions of zero frequency (i.e., a static load) and the phase lead/lag of the response is zero. For these conditions, the equation is responding as a massless elastic system. The time derivative term is negligibly small in this case, so it can be dropped. The resulting expression is the pure elastic representation of the mode:

$$M_{gj} W_j^2 Q_j = G_{gj} \qquad (B8.15)$$

This representation is quite suitable for modes whose frequency is reasonably higher than excitation frequencies of interest (the interest to be supplied by the analyst).

The massless elastic equation can also be written

$$Q_j = G_{gj} / (M_{gj} W_j^2) \qquad (B8.16)$$

which shows that the response becomes less and less as the frequency goes up. This is the rational basis for discarding all high frequency modes in the analysis, beyond the frequency range of interest. This assumption can be made with confidence (assuming that the frequency band of interest in the forcing function has been well established) because the modal participation in the shape of BRL is seen to disappear for the higher modes. Other analysis methods (such as those that use so-called "primitive" modeshapes- shape functions that approximate the shapes of interest but which are not normal modes and therefore do not allow decoupling of the infinite number of equations) can not use this rational basis for reducing the equation set down to a reasonable size.

By restricting the number of normal mode equations only to those modes of interest, the complexity of the forced motion problem has been minimized for a given desired level of fidelity. The numerical processes that are used to solve the equations can be of minimum sophistication, which almost invariably leads to the lowest cost and lowest risk solution of the forced motion problem.

The retained equations can be expressed now in finite form using column vectors to contain the modal coordinates-

$$q^{\cdot\cdot} + E^2_q \cdot q = g \qquad (B8.17)$$

$$g = M^{-1}_g \cdot \int_o^R D^T_\cdot \, p_n \, dS \qquad (B8.18)$$

The array E^2_q. is a diagonal matrix with the eigenvalues placed to correspond to the selected normal coordinates. The array M_g^{-1}. is the diagonal matrix of generalized masses. The column g contains the generalized forcing functions each divided by the associated generalized mass (as indicated by premultiplication by the inverse of the generalized mass matrix).

The matrix D is called the modal matrix. It is composed of all of the eigenfunctions associated with the selected modes. Using this array, all equations B8.10 can be combined into a single matrix expression-

$$w_e \ (S, T) = D(S) . q(T) \qquad (B8.19)$$

where it has been shown that D is a function of S but not of time.

B.9 Geometric Considerations

Equation B8.19 expresses the elastic deflection of the BRL as a function of the modal coordinates, q. If an adequate number of normal modes is included in the model, this expression will be accurate. Experience has shown, however, that BAX motions in a radial direction are extremely sensitive. Very small motions in this direction ultimately cause large loads effects at the blade shanks due to the strong centripetal acceleration field in which the blade is operating.

When a small number of modes is incorporated in the model, Equation B8.19 fails to produce good results for this sensitive radial motion of the BRL for operating conditions where the normal coordinates, q, become large. An alternative method for expressing the BRL position is needed; a method that will produce good results with a small number of modes operating with relatively large excursions.

Such a model can be derived if the BRL is assumed not to stretch under loading. This has been found to be a very good approximation for most rotors. The tensile deformations of the blades is almost invariably negligible.

If the BRL cannot stretch, then a vector of length dS can be considered which projects from the point P (see Figure B.1) in the negative number-1 BAX direction. The vector dS can be resolved by the R. array of direction cosines, to calculate the corresponding vector, db.

$$db = -R^T.i_1 dS$$

where the unit vector has been incorporated to show that dS lies along the number-1 BAX system axis.

Dividing this expression through by dS, and using a prime to denote differentiation with respect to S yields the result-

$$b' = -R^T.i_1 \tag{B9.1}$$

This expression can be integrated with respect to S at any instant of time to produce the BRL positional shape vector, b.

Returning to Equation B8.19, the modal positional and rotational deflections can be expressed-

$$e_b = D_b.q \tag{B9.2}$$

$$e_a = D_a.q \tag{B9.3}$$

The modal matrix D. has simply been separated into two parts, one producing translational deflection and the other producing rotational deflection. Although Equation B9.2 is fully representative of BRL elastic translational position only for small q, experience has found that Expression B9.3 is suitable for representing the angular deflections even for relatively large modal participations.

Equation B9.1 can be used to relate the positional and rotational modal matrices, $D_b.$ and $D_a..$ To do this, consider very small modal motions when the blade is in its quiescent state. Under these conditions the BRL positional shape can be expressed-

$$b = b_{oo} + D_b.q \qquad \text{(B9.4)}$$

Equation B9.1 can be written in expanded form, noting that the array $R_f.$ (due to feathering) is unity for this quiescent consideration-

$$b' = -R^T_{oo}.R^T_e.i_1 \qquad \text{(B9.5)}$$

For small q, the direction cosine matrix due to elastic motion can be expressed in the form-

$$R_e. = I. - X(e_a). \qquad \text{(B9.6)}$$

and from the definition of the quiescent position-

$$b'_{oo} = -R^T_{oo}.i_1 \qquad \text{(B9.7)}$$

Equation B9.4 can be differentiated with respect to S and then combined with Expressions B9.3, B9.5, B9.6 and B9.7 to produce the relationship-

$$D'_b. = R^T_{oo}.X_1(1).D_a. \qquad \text{(B9.8)}$$

This equation can be used to calculate elements in the modal matrices, given other elements, so that the input data to a computer analysis is minimized and made convenient for the user. The expression will also be used later when the rotor model is reformulated to remove small differences of large numbers.

A word of caution is extended with regard to Equation B9.6. This form should be used only for very small q. Significant errors will arise if this form is used with large modal participations (large q) because the approximation does not produce accurate results for the small movements the BRL makes radially with relatively large q. These very small radial movements produce very significant loads due to the strong centripetal acceleration field in which the blade is operating. Because of this, a more rigorous formulation of $R_e.$ is needed in most analyses where q can become large.

Returning to the vector $-dSi_1$, the derivations in Appendix A can be used to calculate the rate of a point on the BRL dS away from the point P. If the difference between the rates of the point P and a point dS away is taken, and the result divided by dS, an expression for rate change vs S is created. Resolving this result to SAX produces an expression for the time derivative of b, differentiated with respect to S-

$$b^{\cdot\prime} = -R^T_{\cdot}\, X(w_{bs}).\, i_1 \qquad \text{(B9.9)}$$

The Coriolis theorem derived in Appendix A can be used to define the second time derivative of b, also differentiated with respect to S-

$$b^{\bullet\bullet\prime} = -R^{T}_{\bullet}\,(X^{2}(w_{bs}). + X(w_{bs}^{\bullet}).)i_{1} \qquad \text{(B9.10)}$$

Equations B9.9 and B9.10 can be integrated from 0 to any position S along the BRL, to produce the first and second time derivatives of the vector b (time derivatives taken with respect to SAX). These entities are essential for substitution into Equations B6.3 and B7.1 in order to solve for the inertial and aerodynamic distributed loads.

The methods of Appendix A can be used to produce the array, R_e. from the Euler angles in the column e_a. These angles are small, but care must be taken in approximating the trigonometric functions comprising R_e. if important effects are not to be lost. Experience suggests that all of the trigonometric functions can be represented as series including all quadratic terms. Cubic and higher order terms can usually be discarded. Note, especially, that the cosine of any of the Euler angles cannot be represented as unity: it must be incorporated as unity less half the angle squared.

Appendix A also shows how to relate the rates resolved to a body-fixed system as functions of the time derivatives of the Euler angles. In the nomenclature of this section, this expression is-

$$w_{bs} = W_{y}.\, e_{a}^{\bullet} \qquad \text{(B9.11)}$$

so that the BAX rate w_{bs} can be computed for substitution into Equation B9.9. Direct differentiation of this expression yields-

$$w_{bs}^{\bullet} = W_{y}^{\bullet}.\, e_{a}^{\bullet} + W_{y}.\, e_{a}^{\bullet\bullet} \qquad \text{(B9.12)}$$

which enables computation of $w_{bs}^{\bullet}$ for substitution into Expression B9.10.

These BAX rotational rate and rate-derivative entities relate BAX motions to SAX. The inertial distributed loads Expressions B6.1 and B6.2 require BAX rotational entities with respect to inertial axes. The methods of Appendix A enable representations of these entities to be expressed as-

$$w_{b} = R.\, w_{s} + w_{bs} \qquad \text{(B9.13)}$$

$$w_{b}^{\bullet} = R.\, w_{s}^{\bullet} + X(w_{b}).\, w_{bs} + w_{bs}^{\bullet} \qquad \text{(B9.14)}$$

B.10 Shank Loads

The equations for distributed inertial and aerodynamic loads can be integrated along the BRL to calculate the loads applied to SAX by the blade. The expressions are-

$$f_{s} = \int_{0}^{R} (f_{i} + f_{a})\, dS \qquad \text{(B10.1)}$$

$$m_{s} = \int_{0}^{R} [X(b).\, (f_{i} + f_{a}) + m_{i} + m_{a}]\, dS \qquad \text{(B10.2)}$$

If it is necessary to calculate the internal loads within the blade, they can be produced by similar equations. One convenient formulation is to integrate the distributed loads starting at the blade tip, so that the initial values of the force and moment vectors supported internally by the

$$f_b = -\int_R^S (f_i + f_a)\ dN \qquad \text{(B10.3)}$$

$$m_b = -\int_R^S [X(b) \cdot (f_i + f_a) + m_i + m_a]\ dS \qquad \text{(B10.4)}$$

B.11 Equation Summary

The previous subsections have presented all of the expressions necessary for the analysis of an aeroelastic rotor system. The equations are very general, and can be further processed from this point depending on the type of analysis to be conducted. Obviously, various assumptions can be invoked to simplify the models depending on the accuracy needed.

Before continuing to refine the models, it is convenient at this point to summarize them. Table B.3 collects the equations needed to form a complete model. Note that the equations carry the same identification labels (e.g., B3.1) as when originally derived in this presentation.

B.12 Reformulation

The equations of aeroelastic rotor analysis are notorious for their subtle combinations which, in effect, produce small differences of large numbers that are very important to the accuracy of the analysis when performed by computer. Making simplifying assumptions in the models is perilous because often such simplifications eliminate very important effects.

There are effects in the models which produce very large numbers in the calculations, but these large numbers subtract in later stages of the analysis leaving only small terms which are very important. The results of these characteristics of the model often cause computer implementations to produce very inaccurate results. Even computers with very large word sizes can be swamped by these problems. When the equations are implemented on a computer with limited significant figure accuracy (such as a 16-bit integer digital computer or an analog computer) these problems can completely destroy the accuracy of the solution.

The answer to these problems lies in reformulating the equations in ways that will cause the subtractions of large numbers to occur in the equations before they are programmed. If this is done correctly, the computer can be used to solve directly for the important "small-difference" parameters, which can retain good accuracy because the large masking terms have been eliminated.

The specific ways of reformulation must be selected with thorough knowlege of the physics of the problem- the models and their special idiosyncrasies must be well understood so that the critical areas that produce numerical problems can be identified and reformulated.

One of the most important areas requiring reformulation lies in the expression for the inertial distributed loading, and more specifically, in the expression for the translational acceleration of BAX with respect to SAX. The large terms appear in the models due to the strong centripetal acceleration field established by the spinning of the rotor system.

A "modified" acceleration expression can be stated as follows-

$$u_{mbr} = u_{br} - \ddot{e}_b - 2X(z) . \dot{e}_b - X^2(z) . (b_{oo} + e_b) \qquad \text{(B12.1)}$$

so that modified inertial and aerodynamic distributed forcing functions can be expressed-

$$f_{mi} = - M_b \, u_{mbr} \qquad (c_g \text{ assumed } = 0) \qquad \text{(B12.2)}$$

$$f_{ma} = f_a - f_{ao} \qquad \text{(B12.3)}$$

The term f_{ao} is simply the distributed aerodynamic load defined for the quiescent loading as discussed in Subsection B.4. This function varies only with S (not with time) and is normally input to the analysis.

The terms incorporated in the acceleration expression essentially comprise the linear components of the acceleration due to BRL elastic movements. Because these are the linear components, they can be processed separately from the nonlinear calculations, and carried through the entire analysis including those areas where large terms subtract. These processes will be conducted in the next subsections.

The terms incorporated in Equations B12.1 have been given signs to cause subtractions from u_{br}. Because of this, u_{mbr} will be significantly smaller than u_{br}.

Other specific terms might be incorporated in this inertial reformulation depending on the judgement of the analyst; The expression presented above has produced good results, however, and is therefore recommended as a method for significantly improving the accuracy of an aeroelastic rotor numerical analysis.

The modified acceleration must be processed into the rotor models in essentially two different ways-

* The generalized force and shank loads expressions B8.18, B10.1 and B10.2 must be processed so that the modified distributed loading functions appear in their integrations on S in lieu of the unmodified loading functions.

* A formulation for f_{mi} is needed wherein the subtractions implicit in Equation B12.1 are made properly; that is, in a manner that will not diminish the accuracy of f_{mi} when it is calculated by computer.

The next subsection shows how to process the generalized and shank loads expressions to produce reformulated models for these entities. Following this presentation, Subsection B.14 addresses the process of summing loads produced by all the blades applied to the gimbal ring; this summation also creates small differences of large numbers that need reformulation. Subsection B.15 presents a carefully formulated model for calculating the modified distributed inertial forcing function.

B.13 Generalized and Shank Loads Expression Reformulation

Equations B12.1 through B12.3 can be rearranged to the forms-

$$f_i = f_{mi} - M_b(\ddot{e}_b + 2X(z) . \dot{e}_b + X^2(z) . (b_{oo} + e_b)) \tag{B13.1}$$

$$f_a = f_{ma} + f_{ao} \tag{B13.2}$$

and these versions for the distributed forcing functions can be substituted into the radial integrals needed to calculate the generalized forcing functions which act on the BRL normal mode equations and the shank load integrals. The paragraphs below describe these substitutions on a step- by-step basis, but intermediate versions of the loads expressions being reformulated are not printed for brevity. The reader should be able to reproduce the intermediate expressions of the models as they are reformulated, if the steps described for the processes are executed carefully.

Some convenient definitions of scalar entities involving the variation of rotor spin rate from the quiescent value are expressed-

$$Z_p = Z / Z_o - 1 \tag{B13.3}$$

$$Z_z = (Z / Z_o)^2 - 1 \tag{B13.4}$$

With these definitions, equation B13.1 can be rearranged as follows-

$$\begin{aligned} f_i = f_{mi} &- M_b[e_b^{\cdot\cdot} + 2X(z_o).\ e_b^{\cdot} + X^2(z_o).\ (b_{oo} + e_b)] \\ &- M_b[-2Z_p\ Z_o\ X_3(1).\ e_b^{\cdot} - Z_z Z_o^2\ (I.-I_3.)\ (b_{oo} + e_b)] \end{aligned} \tag{B13.5}$$

where some of the special characteristics for the cross-product operator, X, (described in Appendix A) have been invoked.

The first bracketed term in Equation B13.5 is the linear portion of f_i that occurs when the BRL is in its quiescent position. The portions of this term that involve acceleration and position, $e_b^{\cdot\cdot}$ and e_b give rise to the mass and stiffness arrays M. and K_I. that were discussed in Subsection B.8. Also note the steady term involving b_{oo} which is part of the quiescent loading function, p_o, that was discussed in Subsection B.4. Terms in the generalized function that emerge because of these effects must be discarded, since these influences have already been subtracted from the nonlinear distributed loads Function B8.3. This process will be described in more detail, below.

Equation B13.5 can be substituted into B8.3 and that result substituted into Equation B8.18 to produce an expression for the generalized forcing function column, g. When this is done a number of constant factors involving integrations with respect to S emerge. These factors have been given special nomenclature and summarized in Table B.4. Using these definitions for the constant operators, the expression for the generalized forcing function can now be written in terms of the modified distributed loading functions-

$$\begin{aligned} g = &\int_o^R G_p.\ (f_{mi} + f_{ma})\ dS \\ &+ (Z_p\ G_{zv}. + G_v.)q^{\cdot} + Z_z\ (g_{bo} + G_{zq}.\ q) + G_k\ (f,\ q).\ q \end{aligned} \tag{B13.6}$$

Equation B9.2 and its time derivatives have been used in the substitutions which produce Expression B13.6.

The operator G_p. which appears in the radial integration is defined in Table B.6 as an aggregate function of S. Its definition in this manner allows a computer solution external to the time-domain solution of the rotor equations to solve for this operator. The resulting operator becomes input data for the time-domain rotor analysis, which lessens the workload in the time domain and therefore increases the computational speed.

When Z_z is zero the rotor spin rate is at its quiescent value. Note that, when this is the case, Equation B13.6 has no linear dependencies on $q^{\cdot\cdot}$ or q. This is because such linear dependencies are presumed purged from this expression, having been included in the eigenanalysis portion of the model. Also note that no constant terms appear in the equation, since these vanish due to the definition of the quiescent loading function, p_o, and the subtraction of this steady column in Equation B8.3.

The stiffness factor, G_k that appears in the equation is defined in Table B.7. This factor arises because of the nonlinear portions of the BRL stiffness influence function addressed in Subsection B.8. Although Table B.7 defines this function, external analyses often produce this G- array directly. The array can be presented in tabular form as a function of f and q. Since this is a stiffness array, it can be effectively included in the equations as variations in the eigenvalue matrix, E_q^2. (See Equation B8.17).

For most analyses, the definition of the BRL quiescent conditions made previously allow this nonlinear stiffness matrix to be neglected. If data is available to describe this array, however, it can be incorporated in the analysis as shown above.

Equation B13.6 combined with Expression B8.17 produces the full blade-motion equation required for the analysis. Only weak functions of q and $\dot{q}$ appear on the right hand side of the diagonalized equation which enhances the stability of a numerical integration process to a considerable degree. Because the left hand side portion of this full blade-motion equation is diagonal, an exact solution for this portion of the equation is easy to formulate for the numerical integration step size on time. Experience has been very good with solving this equation formulated as expressed, above.

The shank force calculation defined by Equation B10.1 can be processed in the same way as the generalized forcing function described above. When the modified distributed loading columns are substituted, constant radial integral operators appear which have been summarized in Table B.4. Unlike the generalized forcing function reformulation, however, the terms associated with quiescent linear acceleration and elastic position effects remain. The term associated with $\ddot{q}$ that remains is particularly problematic: it involves very significant differencing of large terms which can introduce significant errors in the calculation of the shank loads if not reformulated.

The blade motion equation can be used to eliminate $\ddot{q}$ from the shank loads expression. Equation B8.17 produces $\ddot{q}$ while Formula B13.7 produces the generalized forcing function. Using this approach to eliminate $\ddot{q}$ produces the reformulated expression for the shank force-

$$f_s = f_r + F_{sqd}.(\dot{q}/N_b) + F_{sq}.(q/N_b)$$
$$+ f_{so}/N_b + F_{sk}(f,q).(q/N_b) \qquad \text{(B13.7)}$$

where the radial integral portion of the expression is given by-

$$f_r = \int_o^R F_p.(f_{mi} + f_{ma})\, dS \qquad \text{(B13.8)}$$

and where the following define operators that are functions of rotor spin speed-

$$F_{sqd}\cdot = F_{qd}\cdot + F_{zqd}\cdot Z_p \qquad \text{(B13.9)}$$

$$F_{sq}\cdot = F_q\cdot + F_{zq}\cdot Z_z \qquad \text{(B13.10)}$$

$$f_{so} = f_z Z_z + f_o \qquad \text{(B13.11)}$$

The operator F_p. is an aggregate operator that is a function of S. It is defined in Table B.6 in terms of previously-defined entities. The two terms shown for F_p. in Table B.6 incorporate some very small differences of large constants. These differences can be taken in a computer calculation external to that associated with solving the rotor models as functions of time. The calculations can be made using multiple-precision numerics to retain accuracy. Since the subtractions occur before being used in the time-domain calculations, accuracy in the calculation of the shank force can now be retained, even using computers with limited word sizes.

Equation B10.2 for the shank moment can be processed as was the force to yield the expression-

$$m_s = m_r + M_{sqd}.(q^{\cdot}/N_b) + M_{sq}.(q/N_b)$$
$$+ m_{so}/N_b + M_{sk}(f,q).(q/N_b) \qquad \text{(B13.12)}$$

where the radial integral is defined-

$$m_r = \int_o^R [M_{p.}(f_{mi} + f_{ma}) + X(b_p).(f_i + M_b X^2(z).\, b_{oo} + f_a - f_{ao})$$
$$+ X(b_p - e_b).(-M_b X^2(z).b_{oo} + f_{ao}) + m_i + m_a]\; dS \qquad \text{(B13.13)}$$

A "perturbation" value for the BRL position vector is defined

$$b_p = b - b_{oo} \qquad \text{(B13.14)}$$

and special operators that vary with rotor spin rate are defined

$$M_{sq}\cdot = M_q\cdot + M_{zq}\cdot Z_z \qquad \text{(B13.15)}$$

$$M_{sqd}\cdot = M_{qd}\cdot + M_{zqd}\cdot Z_p \qquad \text{(B13.16)}$$

$$m_{so} = m_z Z_z + m_o \qquad \text{(B13.17)}$$

Note that Equation B13.13 shows the unmodified distributed forcing functions being operated on by the cross-product operator $X(b_p)$.. Since this perturbation operator is small, it is not necessary (and it is very inconvenient) to use the modified distributed loading functions in this case. The modified loads are used in the major portion of the integration with operator M_p..

The operator M_p. is an aggragate function of S defined in Table B.6. Like F_p., this operator involves some very small differences of large numbers that are essential to the accuracy of the shank moment calculation. These subtractions are one of the major reasons for reformulating the equations, because most computer solutions will suffer with considerable losses in accuracy if the subtractions occur as the models are being solved in the time domain.

B.14 Multiblade Combinations

The previous subsection addressed the loads calculations, showing how to reformulate the radial integrations which produce loads entities to cause subtractions of large numbers to occur in a computation external to the time-domain solutions.' This reformulation assures accurate calculations for these radial integrations.

As stated previosly, other problematic differencing occurs when the loads produced by all the blades in a rotor are summed at the gimbal ring. These summations, if performed on the shank loads produced for each blade i, $i=1,2,\ldots N_b$, can be expressed as follows-

$$f_g = \sum_i R_3\,(Y_{bi})\cdot f_{si} \tag{B14.1}$$

$$m_g = \sum_i R_3\,(Y_{bi})\cdot m_{si} \tag{B14.2}$$

Significant increases in accuracy can be achieved by defining additional entities called "multiblade coordinates". These entities are simply transformed versions of the normal coordinate column, q, and its time derivative, as if these were loads. The definitions of the multiblade coordinate entities are listed below:

$$q_s = (1/N_b)\sum_i q_i \sin Y_{bi} \tag{B14.3}$$

$$q_c = (1/N_b)\sum_i q_i \cos Y_{bi} \tag{B14.4}$$

$$q_o = (1/N_b)\sum_i q_i \tag{B14.5}$$

$$q_{ds} = (1/N_b)\sum_i q_i^{\bullet} \sin Y_{bi} \tag{B14.6}$$

$$q_{dc} = (1/N_b)\sum_i q_i^{\bullet} \cos Y_{bi} \tag{B14.7}$$

$$q_{do} = (1/N_b)\sum_i q_i^{\bullet} \tag{B14.8}$$

where Y_{bi} is the angle between blade number 1 in the rotor and blade number i, measured as a "number-3" axis rotation between the SAX associated with blade 1 and the gimbal axis system (GAX). Recall that GAX are SAX for blade number 1 in the rotor.

For convenience, the multiblade vectors can be grouped into composite vectors as follows-

$$q_m = (q_o^T,\ q_c^T,\ q_s^T)^T \tag{B14.9}$$

$$q_{md} = (q_{do}^T,\ q_{dc}^T,\ q_{ds}^T)^T \tag{B14.10}$$

Now define a process that takes an array, say A_m. (which is an operator on q or $q^{\bullet}$), and converts it into another form:

$$A_m (A.). = \begin{bmatrix} 0 & A_1 & A_2 \\ 0 & A_2 & -A_1 \\ A_3 & 0 & 0 \end{bmatrix} \qquad \text{Note: } A_i = \text{Row i of A.} \qquad (B14.11)$$

The resulting entity will be called a "multiblade operator", herein.

With these definitions, Equations B14.1 and B14.2 can be stated in a different form which will produce good numerical accuracy when solved in the time domain:

$$f_g = \sum_i R_3(Y_{bi}).f_{ri} + A_m(F_{sq}.).\ q_m + A_m(F_{sqd}.).\ q_{md} + I_3.f_{so} \qquad (B14.12)$$

$$m_g = \sum_i R_3(Y_{bi}).m_{ri} + A_m(M_{sq}.).\ q_m + A_m(M_{sqd}.).\ q_{md} + I_3.m_{so} \qquad (B14.13)$$

These formulations have caused the subtractions among q and q˙ to occur among all the blades in the rotor first, before being operated on by loads-producing arrays F_s. and M_s.. Additionally, subtractions of large steady loads in f_{so} and m_{so} have been eliminated from the model. These include, among other contributors, the steady root tensile force supported by each blade which is a very large force that will mask small but very important "differential tension" terms in almost any type of computer analysis.

Equations B14.12 and B14.13 are the reformulated total gimbal ring loads produced by all blades in the rotor. These will be transformed by the gimbal analysis of Appendix C to produce the final rotor loads applied to the rotor support system.

B.15 Reformation of the Modified Inertial Acceleration

An expression is needed for u_{mbr} that will produce an acceptably-accurate calculation for the modified distributed inertial forcing function, f_{mi}, which is used for many radial loads integrations. Equation B12.1 presents the basic definition of this modified acceleration, but this expression cannot be used as-is because small differences of large numbers will destroy its accuracy. It is necessary to decompose specific elements of the total acceleration vector, u_{br}, and then subtract the necessary associated entities (term by term) so that the differencing produces acceptable results.

To do this, define "deflection vectors" as follows-

$$x = b - b_{oo} - e_b \qquad (B15.1)$$

$$x^{\cdot} = b^{\cdot} - e_b^{\cdot} \qquad (B15.2)$$

$$x^{\cdot\cdot} = b^{\cdot\cdot} - e_b^{\cdot\cdot} \qquad (B15.3)$$

These vectors are the differences between the actual BRL position (and its time derivatives) and the BRL position that would be predicted by the linear model of Equation B9.2 (and its time derivatives). Using these deflection vectors, Equation B12.1 can be rearranged to the form-

$$u_{mbr} = u_s + x^{\cdot\cdot} + 2X(w_s^{\cdot} - z).b^{\cdot} + 2X(z).x^{\cdot} + X(w_s^{\cdot}).b$$
$$+ [X(w_s - z).X(w_s). + X(z).X(w_s - z).]b + X^2(z).x \tag{B15.4}$$

In addition to the incorporation of the deflection vectors in critical areas involving subtractions, the rotor spin vector, z, has been subtracted from the rotational velocity of SAX (w_s) in critical areas, so that accuracy will be preserved when u_{mbr} is created numerically in a computer.

Expressions B15.1 through B15.3 defining the deflection vector (and its time derivatives) can be differentiated with respect to S, yielding terms (with primes) that have been defined in previous sections of this document. Equations B9.1, B9.9 and B9.10 can be used for derivatives of the BRL position vector, b. Equations B9.2 and B9.8 yield expressions for the derivatives of e_b. Equation B9.7 provides an expression for the quiescent BRL positional shape.

These definitions produce complete formulations for the deflection vector and its time derivatives. In a computer analysis, the tensor equations presented herein need to be expanded into scalar form. When this is done, a number of important subtractions occur which eliminate the small differences of large numbers that cause numerical problems. These subtractions are best performed when the expressions are in their scalar forms; the tensor definitions to not lend themselves to convenient variations which allow the subtractions to be made explicitly.

The deflection vectors and their time derivatives are finally produced in an analysis by integrating their primed counterparts with respect to S.

B.16 Collection, Reorganization and Summary of Rotor Models

The previous subsections of this document have presented derivations for a general aeroelastic rotor model; the derivations have followed the physical phenomena which give rise to the various elements of the model.

This subsection deals with collecting, reorganizing and summarizing the models for convenient implementation in a computer program. Figure B.3 is a signal flow chart showing "subsystems" of the rotor analysis, each containing some of the math models previously derived. Lines among the blocks carry nomenclature representing variables input to and output from the various portions of the analysis. The analysis subsystems (or model groups) have been given names and numbers, for convenience. Figure B-4 shows a flow diagram of the computations in the rotor model.

The nomenclature presented by Figure B.3 closely resembles the engineering nomenclature appearing in previous subsections and summarized by Table B.1. All the letters are upper case, however, and super/subscripts have been eliminated. In this form the nomenclature is suitable for use in actual computer code.

Some key conventions have been followed in converting the engineering notation into computer notation. The nomenclature label always begins with a letter. Characters following this lead letter character refer first to superscripts and then to subscripts. Finally, numeral characters define specific elements of vectors (one-dimensional arrays) or operators (two-dimensional arrays). Note that the "dots" on the second-order arrays in the engineering analysis are dropped in the computer nomenclature, being replaced by two numeral subscripts.

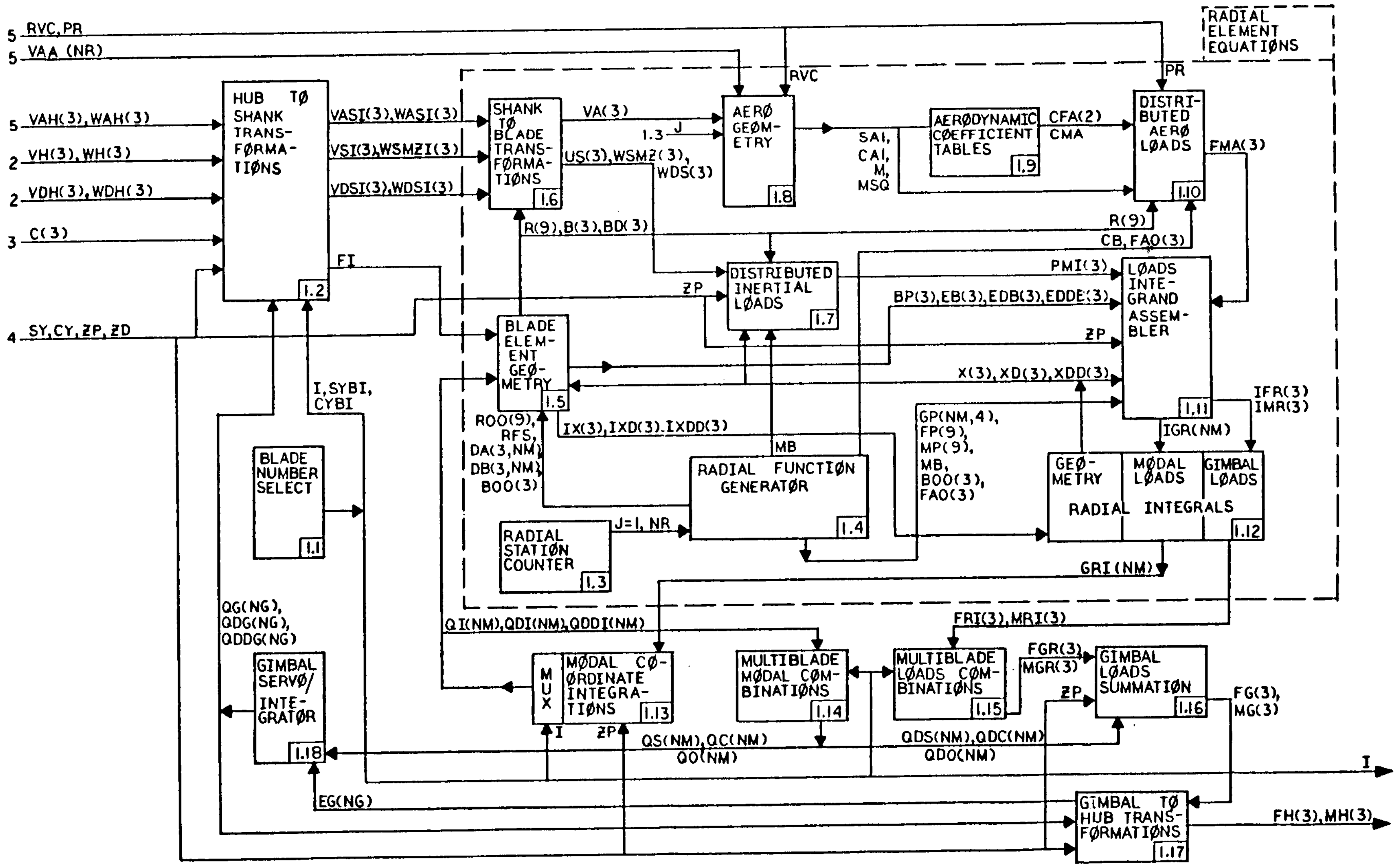
In addition to these conventions, "primed" variables that are to be integrated with respect to S have been given "I" lead characters for "integrand". The remainder of these labels follows the other conventions listed above.

Two-dimensional arrays are to be processed by the computer as "packed" arrays. The convention used herein is to pack by column, as is standard in the Fortran language. In these cases, the array is indicated with a single number in parentheses, where the number represents the total number of elements in the array.

For example, the operator R. is represented in the computer variables as R(9), which is a string of nine elements of R stored as three columns.

As stated above, the models developed in previous subsections of this document need rearranging and collecting for computer implementation. Tables B.8.1-18 present the models so processed, appearing still in engineering notation, but with some changes. These changes (made to aid the computer implementation) will be described in the following subsections. Key assumptions incorporated in the analysis at this point are underlined. The subsection numbers are keyed to the subsystem numbers of Figure B.3.

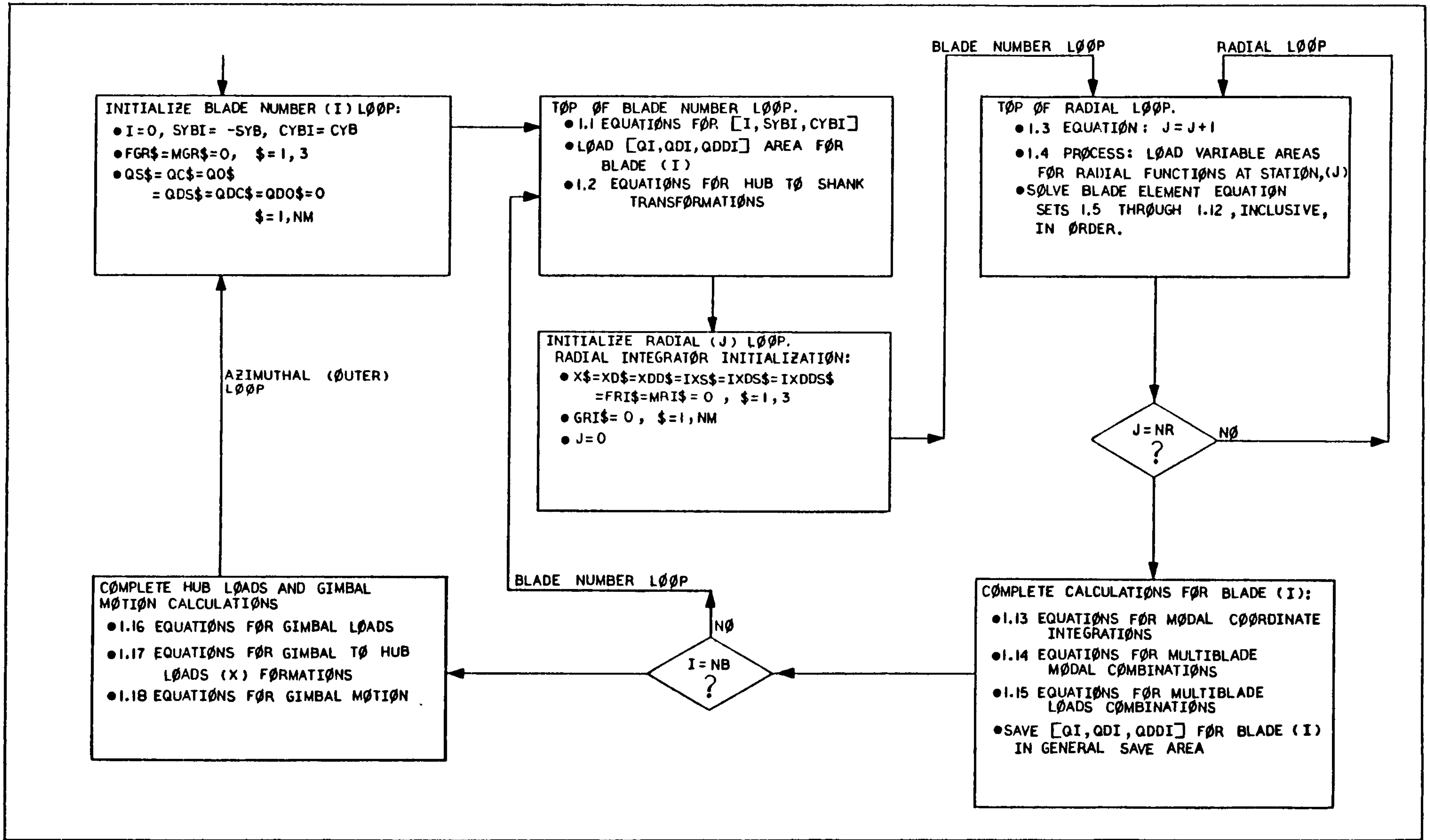
The mathematical model of the rotor has been expanded into scalar equations and presented in Appendix F; these are the "physical equations". The equations are presented under numerical subheadings which key to the subsystem block numbers of Figure B.3.


RADIAL ELEMENT EQUATIØNS
5 RVC, PR
5 VAA (NR)
5 VAH(3), WAH(3)
2 VH(3), WH(3)
2 VDH(3), WDH(3)
3 C(3)
4 SY, CY, ZP, ZD
HUB TØ SHANK TRANS-FØRMA-TIØNS 1.2
VASI(3), WASI(3)
VSI(3), WSMZI(3)
VDSI(3), WDSI(3)
FI
I, SYBI, CYBI
BLADE NUMBER SELECT 1.1
SHANK TØ BLADE TRANS-FØRMA-TIØNS 1.6
VA(3)
1.3 J
US(3), WSMZ(3), WDS(3)
R(9), B(3), BD(3)
RVC
AERØ GEØM-ETRY 1.8
SAI, CAI, M, MSQ
AERØDYNAMIC CØEFFICIENT TABLES 1.9
CFA(2) CMA
PR
DISTRI-BUTED AERØ LØADS 1.10
FMA(3)
R(9)
CB, FAO(3)
DISTRIBUTED INERTIAL LØADS 1.7
ZP
PMI(3)
BP(3), EB(3), EDB(3), EDDE(3)
ZP
X(3), XD(3), XDD(3)
LØADS INTE-GRAND ASSEM-BLER 1.11
IFR(3) IMR(3)
IGR(NM)
BLADE ELEM-ENT GEØ-METRY 1.5
ROO(9), RFS, DA(3,NM), DB(3,NM), BOO(3)
IX(3), IXD(3), IXDD(3)
MB
RADIAL FUNCTIØN GENERATØR 1.4
GP(NM,4), FP(9), MP(9), MB, BOO(3), FAO(3)
GEØ-METRY
MØDAL LØADS
GIMBAL LØADS
RADIAL INTEGRALS 1.12
RADIAL STATIØN COUNTER 1.3
J=1, NR
GRI(NM)
QG(NG), QDG(NG), QDDG(NG)
GIMBAL SERVØ/ INTE-GRATØR 1.18
QI(NM), QDI(NM), QDDI(NM)
M U X
MØDAL CØ-ØRDINATE INTEGRA-TIØNS 1.13
I
ZP
QS(NM), QC(NM) QO(NM)
MULTIBLADE MØDAL CØM-BINATIØNS 1.14
FRI(3), MRI(3)
MULTIBLADE LØADS CØM-BINATIØNS 1.15
FGR(3) MGR(3)
ZP
GIMBAL LØADS SUMMATIØN 1.16
FG(3), MG(3)
QDS(NM), QDC(NM) QDO(NM)
I
EG(NG)
GIMBAL TØ HUB TRANS-FØRMATIØNS 1.17
FH(3), MH(3)

RØTØR MØDEL-
SIGNAL FLØW DIAGRAM

FIGURE B.3

REV:


INITIALIZE BLADE NUMBER (I) LØØP:
● I=0, SYBI= -SYB, CYBI= CYB
● FGR$=MGR$=0, $=1, 3
● QS$=QC$=QO$ = QDS$=QDC$=QDO$=0 $=1,NM
TØP ØF BLADE NUMBER LØØP.
● 1.1 EQUATIØNS FØR [I, SYBI, CYBI]
● LØAD [QI, QDI, QDDI] AREA FØR BLADE (I)
● 1.2 EQUATIØNS FØR HUB TØ SHANK TRANSFØRMATIØNS
INITIALIZE RADIAL (J) LØØP.
RADIAL INTEGRATØR INITIALIZATIØN:
● X$=XD$=XDD$=IXS$=IXDS$=IXDDS$ =FRI$=MRI$=0 , $=1, 3
● GRI$= 0, $=1, NM
● J=0
BLADE NUMBER LØØP
RADIAL LØØP
TØP ØF RADIAL LØØP.
● 1.3 EQUATIØN: J=J+1
● 1.4 PRØCESS: LØAD VARIABLE AREAS FØR RADIAL FUNCTIØNS AT STATIØN,(J)
● SØLVE BLADE ELEMENT EQUATIØN SETS 1.5 THRØUGH 1.12, INCLUSIVE, IN ØRDER.
J=NR ?
NØ
AZIMUTHAL (ØUTER) LØØP
BLADE NUMBER LØØP
NØ
I=NB ?
COMPLETE HUB LØADS AND GIMBAL MØTIØN CALCULATIØNS
● 1.16 EQUATIØNS FØR GIMBAL LØADS
● 1.17 EQUATIØNS FØR GIMBAL TØ HUB LØADS (X) FØRMATIØNS
● 1.18 EQUATIØNS FØR GIMBAL MØTIØN
CØMPLETE CALCULATIØNS FØR BLADE (I):
● 1.13 EQUATIØNS FØR MØDAL CØØRDINATE INTEGRATIØNS
● 1.14 EQUATIØNS FØR MULTIBLADE MØDAL CØMBINATIØNS
● 1.15 EQUATIØNS FØR MULTIBLADE LØADS CØMBINATIØNS
● SAVE [QI, QDI, QDDI] FØR BLADE (I) IN GENERAL SAVE AREA

RØTØR MØDEL SØFTWARE - CØNTRØL FLØW STRUCTURE

FIGURE B.4

REV:

B.16.1 Blade Number Select

This subsystem produces the blade number of the rotor to be processed in series by counting from 1 to N_b. Additionally, the sine and cosine of the angle rotating GAX to SAX for blade i are computed. Last, the state variables (modal coordinates) for blade i are found in a memory region storing these variables for all blades, and placed in a region to be used in calculations for blade i.

B.16.2 Hub to Shank Transformations

Table B.8.2 summarizes a number of coordinate transformations used in the model. An important feature incorporated in Table B.8.2 should be noted - the subtraction of the rotor spin vector, z, from the inertial velocity columns. This has been done to improve the scaling in the inertial velocity processing further downstream in the analysis. The vector, z, is simply subtracted from all of the rotational inertial velocity entities, and the equations modified as required for consistency throughout. The parameter WSMZI(3) appears in Figure B.3 representing w_{si}-z produced by the inertial velocity calculations, so that z does not mask small values in w_{si}.

B.16.3 Radial Station Counter

This subsystem simply produces the parameter J counting from 1 to NR, the number of radial stations used in the blade integrations on S. Radial stations are those places along the BRL where physical parameters are defined. Each set of physical parameters is identified with a specific S value, for use in the integrations. Although not strictly necessary, it is assumed that blade radial stations are equi-spaced in the rearranged rotor analysis.

B.16.4 Radial Function Generator

This subsystem moves the physical data that is a function of S from the complete data table to a smaller string to apply only to the present station being analyzed, station J. This block transfer is made all at once to avoid later repetitive use of indirect addressing to fetch these parameters: addressing that is slower than immediate addressing which is possible when all parameters for a station are consolidated in a physical memory address space.

B.16.5 Blade Element Geometry

Expressions from previous derivations are collected for this geometry section. The equation numbers associated with the models are listed. Sometimes more than one equation number is listed, which indicates that the noted expressions were rearranged to produce the result listed in Table B.8.5. In most cases these rearrangements are straightforward and require no additional amplification here.

Note that some definitions have been made in Table B.8.5. One is an assumption on the form of the aray R_F.. Firstly, it is assumed that there is only one mechanical feathering control parameter, f=F. Additionally, R_F. is assumed to be a pure rotation about the number-1 BAX axis to move the BRL from the quiescent shape to the control shape. This is a good assumption for most rotor systems.

It has been assumed that $\dot{W}_y$. has negligible effect, so it has been dropped from Equation B9.12. This term would fall out later in the analysis anyway (where cubic and higher order terms in series representations of trig functions are assumed negligible).

A series approximation for sine and cosine of the feathering angle has been made (equations for F_s) which is very accurate for feathering excursions between +-90 degrees. Rotorcraft seldom see such excursions, but propellers and wind turbines do.

B.16.6 Shank to Blade Transformations

Equations are gathered from previous derivations as indicated in Table B.8.6. The equations are repeated without modification.

B.16.7 Distributed Inertial Loads

The inertial moment applied to the BRL is assumed zero in this collection; this is done because the inertial twisting moments are invariably small as they influence loads applied to SAX. These moments are significant in the blade torsional equations, but it is assumed here that these significant effects have been relegated to the eigenanalysis portion of the torsional analysis. In other words nonlinear inertial twisting moments have been neglected.

B.16.8 Aero Geometry

The aero geometry equations have been rearranged to appear in components of Mach number. The parameter $R_{vc} = 1/R_c$ (the reciprocal of sonic velocity) has been introduced to eliminate the need for division in the time-domain solutions, so that components of velocity multiplied by this factor yield components of mach number for the calculations.

B.16.9 Aerodynamic Coefficient Tables

As before, it is assumed that such tables and proper table look-up routines are available for the analysis. Spanwise force and number-2 and number-3 axis aerodynamic moments have been neglected as shown in Table B.8.9.

B.16.10 Distributed Aerodynamic Loads

Equations have been gathered from previous analyses, as shown. The modified loads column, p_{ma}, is shown composed of the modified force and the unmodified moment vectors.

B.16.11 Loads Integrand Assembler

For ease in the computer analysis, the integrands in the loads radial integral operations have been given special "primed" notation in this subsection. Refer to the equations with numbers shown in Table B.8.11 to see the origins of these integrand expressions. When converted to computer nomenclature, these items will have an "I" lead character to denote that they are integrands for radial integration.

B.16.12 Radial Integrations

These models simply indicate radial integration of those "primed" integrands as required to produce final loads and deflection vector entities.

B.16.13 Modal Coordinate Integrations

The blade motion equation is numerically integrated to calculate the "q" variables as functions of time. As discussed previously, the nonlinear portion of the model (now included in the generalized forcing function vector, g, which was produced by radial integration) is held constant for an interval of time associated with the numerical integration time step size. The linear portion of the equation (Expression B8.17) is solved exactly over the interval. This process provides for exact solution of the powerful mass and stiffness phenomina, while the inexact (transport-delayed) portion of the numerical solution is restricted to the weaker nonlinear terms. This approach has worked well in past simulation experience with the rotor blade motion equations.

Straightforward methods of ordinary differential equations can be used to find the exact solutions. The values of q and $q^{\cdot}$ are essentially initial conditions which are to be refreshed to new values one time step, say D_t away. Table B.8.13 lists the equations required to "move" the variables over the interval. The expressions are in terms of constants involving D_t and the eigenvalues in E_q^2. Expressions for the constants used in the solutions are also given in Table B.8.13. These costants can be easily determined in a preprocessing phase so that they do not impede the performance of the time-domain analysis.

B.16.14 Multiblade Modal Combinations

These models have been taken without modification from the previous derivations as indicated by the equation labels in Table B.8.14.

B.16.15 Multiblade Loads Combinations

Equations B14.12 and B14.13 derived previously have been separated for convenience as shown in Table B.8.15. The summations of f_{ri} and m_{ri} among all the blades in the rotor are made first. These components of the total gimbal loads due to radial integration are then added to the other contributions (from multiblade coordinates) to produce the final gimbal loads summation. A number of operators must be calculated that are functions of rotor spin rate. The necessary expressions are shown, taken directly from previous analyses without modification.

B.16.16 Gimbal Loads Summation

The expressions of Table B.8.17 are summations of the loads acting on the outer gimbal ring.

B.16.17 Gimbal-To-Hub Transformations (Loads)

The expressions of Table B.8.17 are used in the gimbal analysis presented in Appendix C.

B.16.18 Gimbal Servo Integrator

Appendix D describes the technique used for the gimbal portion of the analysis, where a "gimbal error" (the moment about a hinge that should vanish or have a specific value) is driven to the correct value by a servo approach. The gimbal error is multiplied by a gain (K_{ge}) to become the gimbal acceleration.

As described in Appendix D, the concept works but can encounter stability problems due to the transport delay associated with the numerical integration process. The instability is also involved with the blade modal degrees of freedom to a significant degree.

As discussed in Appendix D, small feedback terms from the multiblade coordinates associated with the blade first flapping degree of freedom stabilized the servo loop and resulted in good performance in reducing the gimbal error to zero. Accordingly, terms have been added to the servo equation of Table B.8.18 which incorporate the necessary multiblade coordinates. The gains in these stabilizing feedback loops are input as constants to the analysis; these have been given special nomenclature in Table B.8.18.

The numerical integration of the gimbal acceleration to produce rate and position is also described by equations in Table B.8.18. The constants in Table B.8.18 can be obtained by an external preprocessing computer.

Table B.1 NOMENCLATURE USED IN THE AEROELASTIC ROTOR ANALYSIS

General Nomenclature Conventions-

u	translational acceleration vector
v	translational velocity
w	rotational velocity
f	force vector
m	moment vector
g	generalized forcing function vector
q	normal coordinate vector
R.	matrix of direction cosines
p	column of loads, force and moment vectors combined

General Subscript conventions

oo	referring to the BRL quiescent shape
o	referring to the BRL control position
a	referring to aerodynamic entities
i	referring to inertial loading entities

Nomenclature definitions

A_b	Angle of rotation between BAX and the local wind, about the the number-1 axis.
A_m.	Multiblade operator producing a force or moment column by operating on a multiblade coordinate vector.
b	Vector locating BAX with respect to SAX, resolved to SAX.
b_p	Perturbation version of b. $b_p = b - b_{oo}$
b_o	The b vector when the BRL is in its "control position".
b_{oo}	The b vector when the BRL is in its quiescent shape.
c_g	Vector locating the blade section center of gravity with respect to BAX, resolved to BAX.
C_b	Scalar: local blade chord at S.

c_{fa}, c_{ma} Aerodynamic coefficients: nondimensionalized force and moment vectors for the blade section at S.

$D.$
$D_a.$
$D_b.$ Modal matrices. D. produces the full six-element deflection vector, w_e, by operating on the modal column, q. $D_a.$ and $D_b.$ produce the angular and positional subvectors of w_e by operating on q; they are submatrices of D.

e_a, e_b Elastic deflections: the first is a column of Euler angles angles describing the rotary deflection of BAX from the control position to the final position. The second column, e_b, is the translational movement of BAX from the control position to the instantaneous (final) position, due to small modal participation, q.

$E_q^2.$ Diagonal matrix of eigenvalues (i.e., natural frequencies, squared) for all the modes included in a finite blade model.

f Force column. Subscripts denote the following:

s	acting on SAX
r	component of f_s due to radial integration
i	distributed force on the BRL due to inertial loading
mi	"modified" inertial distributed forcing function
a	distributed force on the BRL due to aerodynamic loading
ma	"modified" aerodynamic distributed forcing function
ao	aerodynamic distributed force for BRL quiescent shape
g	force on the gimbal ring due to all blades, in GAX

$F_{sq}.$, $F_{sqd}.$, $F_{zq}.$, $F_{zqd}.$, $F_p.$, etc.

Operators in the expression for force applied to SAX.

f_{so}, f_z, f_o Columns in the expression for force applied to SAX.

G_{gj} The generalized forcing function for the jth normal mode of vibration.

g A column containing all the generalized forcing functions incorporated in a finite mode model (all forcing functions divided by their respective generalized masses).

$G_v.$, $G_{zq}.$, $G_{zv}.$, $G_p.$, etc.

Operators in the expression for generalized forcing function.

$I_b.$ The distributed inertial tensor for the blade element at S, resolved to BAX.

$K_e.$, $K_{ep}.$ Kernal function arrays: BRL stiffness functions. The first is the general (nonlinear) function while the second is a linearized version valid only for small deflections of the BRL from its control position.

K_I Distributed blade inertial stiffness matrix that arises due to movement of the blade element mass in the centripetal acceleration field.

m Moment column vector. Subscripts define the specific moment column involved; similar to those for the force column, f.

M_b Scalar blade distributed mass

M Mach number of the blade section at S.

$M.$ Distributed mass matrix which operates on the double time derivative of the BRL deflection vector, w_e, to produce a contribution to the distributed inertial loading column, p_i.

M_{gi} Generalized mass for the ith blade mode.

$M_g.$ Diagonal matrix of generalized masses for all modes included in a finite normal mode model for the blade.

$M_{sq}.$, $M_{sqd}.$, $M_{zq}.$, $M_{zqd}.$, $M_p.$, etc.

Operators in the expression for moment applied to SAX.

m_{so} Column in the expression for applied SAX moment.

N_b Number of blades in the rotor.

p General loads column formed by stacking f and m force and moment column vectors. Subscripts listed under the "f" definition, above, also are used to modify this loads column.

P_r Pressure ratio: ratio of ambient atmospheric pressure to sea level standard pressure (2116 psf).

Q_a Aerodynamic pressure at S.

Q_{ac} Aerodynamic pressure times blade chord dimension at S.

Q_i Normal coordinate for mode i.

q Column of normal coordinates for a normal mode blade model.

q_s, q_c, q_o Multiblade coordinates: sine, cosine and steady entities.

q_{ds},q_{dc},q_{do} Multiblade coordinates for $q^{\cdot}$: sine, cosine and steady components.

q_m, q_{md} Composite multiblade coordinate vectors made by stacking the generic positional and velocity multiblade coordinate vectors.

R Rotor blade radius

R. Rotational matrix of direction cosines. Subscripts are used to further define the matrix by indicating the starting and ending coordinate system in the rotations. Subscripts are used as follows-

oo Rotate from SAX to the quiescent position of BAX.
f Rotate from quiescent position of BAX to control position.
o Rotate from SAX to the control position of BAX.
e Rotate from the control position of BAX to final position.
none Rotate from SAX to final (instantaneous) BAX position.

S Spatial (line) coordinate measured along the BRL from SAX to BAX. Represents the actual (line) span dimension when the the blade is unloaded.

T Time.

u_a Acceleration of SAX with respect to IAX, resolved to SAX.

u_{br} Acceleration of BAX with respect to IAX, resolved to SAX

u_{mbr} Modified version of u_{br} caused by reformulation to eliminate small differences of large numbers.

v Translational velocity vector- subscripts as follows:

b S:BAX,B:IAX,R:BAX
s S:SAX,B:IAX,R:SAX
as S:SAX,B:AIR,R:SAX
ar S:SAX,B:AIR,R:BAX. Assumes air moves as rigid body.
aa S:BAX,B:AIR,R.BAX. Local interference air motion.
a S:BAX,B:AIR,R:BAX. Total local airspeed.

V_c Velocity of sound.

w Rotational velocity vector - subscripts listed under the v translational velocity definition also modify w. Additional subscripts on w are-

bs S:BAX,B:SAX,R:BAX
e Elastic deflection vector (not a rotational velocity vector). It is a generalized translational and rotational position deflection for the BRL moving away from the control position.

w_i Modeshape for the i-th normal mode of vibration.

W_i Natural frequency of vibration for the i-th normal mode.

W_y. Matrix which operates on the time derivatives of Euler angles to produce the rotational velocity of a subject axis system with respect to a base system.

x Deflection vector; the difference between the BRL position predicted by a linear modal matrix and the actual position (instantaneous position): a function of S and T.

X. Cross product operator. X(a).b represents the vector cross product between vectors a and b.

Y_{bi} Scalar angle between blade number i and blade number 1. This is a number-3 axis rotation between GAX and SAX for blade i.

Z, Z_o Rotor spin rate. Subscript o denotes the quiescent value of Z.

z Vector of all zeros except -Z in the number 3 position: the vectorial rotor rotational velocity resolved to HAX.

Table B.2 COORDINATE SYSTEMS USED IN ROTOR ANALYSIS

BAX- Blade Axes

Fixed to a blade element:. origin identified along the BRL by the line (spatial) parameter, S. Number-1 axis points generally towards hub and is tangent to BRL. Number-2 axis related to blade element by analyst, usually parallel to the airfoil chord at S.

BRL- Blade Reference Line

Locus of BAX origins along blade span- these points are considered fixed to mass molecules of the blade sections, and are chosen within each section by the analyst. The section airfoil quarter-chord point is a typical BRL definition. The line parameter, S, measures the distance along the BRL from the SAX origin to a particular BAX location. S is the actual length when the blade is unloaded, but under loading, a particular value of S will select the same BAX as would have been selected by that value of S when the blade is unloaded.

GAX- Gimbal Axes

Fixed to outer gimbal ring so that the number-3 axis is coincident with the number-3 axis of HAX when the gimbal is undeflected. The negative number-1 GAX is on or near the projection of the BRL on the number-1,2 GAX plane, for blade number-1 in the rotor. GAX and rotor axes (RAX) are coincident when the gimbal is undeflected. For teetered rotors with a delta-3 angle, GAX is not aligned with the teetering axes (see the definition of TAX, and also Appendix C)

HAX- Hub axes

This frame is fixed to the support system of the rotor so that its number-3 axis lies coincident with the rotor shaft. The origin is inside the hub as near as possible to the closest point between the BRL and the shaft centerline. This HAX origin is the "reference point" for the rotor. Motion entities for HAX are defined to the rotor analysis; loads applied to the supports by the rotor (resolved to HAX) are produced by the rotor analysis.

IAX- Inertial Axes

A Newtonian frame fixed to the stars (i.e., Newton's second law of motion, f=ma, holds in this frame).

RAX- Rotor Axes

The origin is coincident with the HAX origin as are the number-3 axes. RAX spin with the rotor a rate Z about the negative number-3 axis.

SAX- Shank Axes

The origin is coincident with the GAX origin, as are the number-3 axes. SAX are rotated about the negative number-3 axis the angle Y_{bi} so that the negative number-1 axis lies under the BRL projection on the number 1,2 plane for blade number i. SAX for blade number 1 are coincident with GAX.

C-2

Detailed definitions of the following axes systems are given Appendix C.

TAX - Teetering Axes

The origin is coincident with the RAX origin, as are the number-3 axes. TAX are rotated the angle γ_c about the number-3 axis from so that the TAX number-2 axis is parallel to the teetering hinge.

CAX - C Axes

Fixed to the outer gimbal, and coincident with TAX when the gimbal is undeflected.

TUAX, CUAX - Axes system which account for rotor undersling.

Table B.3 FUNDAMENTAL ROTOR MODEL - EQUATION SUMMARY

Geometry

$$R_o \cdot = R_f \cdot R_{oo} \cdot \tag{B3.1}$$

$$R \cdot = R_e \cdot R_o \cdot \tag{B3.2}$$

$$b' = -R^T \cdot i_1 \tag{B9.1}$$

$$e_b = D_b \cdot q \tag{B9.2}$$

$$e_a = D_a \cdot q \tag{B9.3}$$

$$w_e = D \cdot q \tag{B8.19}$$

$$D_b' \cdot = R_{oo}^T \cdot X_1 (1) \cdot D_a \cdot \tag{B9.8}$$

$$b^{\cdot\prime} = -R^T \cdot X (w_{bs}) \cdot i_1 \tag{B9.9}$$

$$b^{\cdot\cdot\prime} = -R^T \cdot (X^2 (w_{bs}) \cdot + X (\dot{w}_{bs}) \cdot) i_1 \tag{B9.10}$$

$$w_{bs} = W_y \cdot \dot{e}_a \tag{B9.11}$$

$$\dot{w}_{bs} = \dot{W}_y \cdot \dot{e}_a + W_y \cdot \ddot{e}_a \tag{B9.12}$$

$$w_b = R \cdot w_s + w_{bs} \tag{B9.13}$$

$$\dot{w}_b = R \cdot \dot{w}_s + X (w_b) \cdot w_{bs} \tag{B9.14}$$

Generalized Forcing Function

$$g = M_g^{-1} \cdot \int_o^R D^T \cdot p_n \, dS \tag{B8.18}$$

Blade Motions

$$\ddot{q} + E_q^2 \cdot q = g \tag{B8.17}$$

Shank Loads

$$f_s = \int_o^R (f_i + f_a) \, dS \tag{B10.1}$$

$$m_s = \int_o^R [X(b) \cdot (f_i + f_a) + m_i + m_a] \, dS \tag{B10.2}$$

Nonlinear Distributed Loads

$$p_n = p_i + M \cdot \ddot{w}_e + K_I \cdot w_e + p_a - p_o$$
$$- \int_o^R (K_e \cdot - K_{ep} \cdot) \, w_e \, dN \tag{B8.3}$$

Table B.3 (continued)

Distributed Inertial Loads

$$f_i = -M_b \; (u_{br} + R^T . (X(\dot{w}_b) . + X^2 \; (w_b) .) \; c_g) \tag{B6.1}$$

$$m_i = -M_b \; X(c_g) . \; R . \; u_{br} + I_b . \dot{w}_b + X(w_b) . I_b . w_b \tag{B6.2}$$

$$u_{br} = u_s + \ddot{b} + 2X(w_s) . \dot{b} + X(\dot{w}_s) . b + X^2 (w_s) . b \tag{B6.3}$$

$$u_s = \dot{v}_s + X \; (w_s) . \; v_s \tag{B6.4}$$

$$p_i = (\; f_i^T \; , \; m_i^T \;)^T \tag{B6.5}$$

Distributed Aerodynamic Loads

$$v_{ar} = R . \; (v_{as} + \dot{b} + X(w_{as}) . b \tag{B7.1}$$

$$v_a = v_{ar} + v_{aa} \tag{B7.2}$$

$$M^2 = (v_{a2}^2 + v_{a3}^2) \; / \; v_c^2 \tag{B7.3}$$

$$A_b = \tan^{-1} \; (v_{a3}/v_{a2}) \tag{B7.4}$$

$$Q_a = 1481 \; M^2 \; P_r \tag{B7.5}$$

$$Q_{ac} = Q_a \; C_b \tag{B7.6}$$

$$c_{fa} = c_{fa} \; (A_b, M) \tag{B7.7}$$

$$c_{ma} = c_{ma} \; (A_b, M) \tag{B7.8}$$

$$f_a = Q_{ac} \; R^T . \; R_1^T \; (A_b) . \; c_{fa} \tag{B7.9}$$

$$m_a = Q_{ac} \; C_b \; c_{ma} \tag{B7.10}$$

$$p_a = (f_a^T, m_a^T)^T \tag{B7.11}$$

Table B.4 CONSTANT OPERATORS FROM RADIAL INTEGRATIONS

Generalized Forcing Function Operators

$$g_{bo} = Z_o^2\, M_{g\cdot}^{-1} \int_o^R M_b\, D_{b\cdot}^T\, (I. - I_3.)\, b_{oo}\, dS$$

$$G_{v\cdot} = -2\, M_{g\cdot}^{-1} \int_o^R M_b\, D_{b\cdot}^T\, X(z_o).\, D_{b\cdot}\, dS$$

$$G_{zv\cdot} = +2Z_o\, M_{g\cdot}^{-1} \int_o^R M_b\, D_{b\cdot}^T\, X(i_3).\, D_{b\cdot}\, dS$$

$$G_{zq\cdot} = +Z_o^2\, M_{g\cdot}^{-1} \int_o^R M_b\, D_{b\cdot}^T\, (I. - I_3.)\, D_{b\cdot}\, dS$$

Shank Force Operators

$$F_{u\cdot} = -\int_o^R M_b\, D_{b\cdot}\, dS$$

$$F_{v\cdot} = -2\int_o^R M_b\, X(z_o).\, D_{b\cdot}\, dS$$

$$F_{zv\cdot} = +2Z_o \int_o^R M_b\, X(i_3).\, D_{b\cdot}\, dS$$

$$F_{b\cdot} = -\int_o^R M_b\, X^2(z_o).\, D_{b\cdot}\, dS$$

$$F_{zb\cdot} = +Z_o^2 \int_o^R M_b\, (I. - I_3.)\, D_{b\cdot}\, dS$$

$$f_{sao} = \int_o^R f_{ao}\, dS$$

$$f_{zo} = +Z_o^2 \int_o^R M_b\, (I. - I_3.)\, b_{oo}\, dS$$

$$f_{io} = -\int_o^R M_b\, X^2(z_o).\, b_{oo}\, dS$$

Shank Moment Operators

$$M_{u\cdot} = -\int_o^R M_b X(b_{oo}).D_{b\cdot}\, dS$$

$$M_{v\cdot} = -2\int_o^R M_b X(b_{oo}).\, X(z_o).D_{b\cdot}\, dS$$

$$M_{zv\cdot} = +2Z_o \int_o^R M_b X(b_{oo}).\, X(i_3).D_{b\cdot}\, dS$$

$$M_{b\cdot} = -\int_o^R [M_b(X(b_{oo}).X^2(z_o)+Z_o^2 X((I. - I_3.)b_{oo}).)+X(f_{ao}).]D_{b\cdot}\, dS$$

$$M_{zb\cdot} = +Z_o^2 \int_o^R M_b[X(b_{oo}).(I. - I_3.) - X((I. - I_3.)b_{oo}).]D_{b\cdot}\, dS$$

$$m_{ao} = \int_o^R X(b_{oo}).f_{ao} dS$$

$$M_{zo} = Z_o^2 \int_o^R M_b X(b_{oo}).(I. - I_3.)b_{oo}\, dS$$

$$M_{io} = -\int_o^R M_b X(b_{oo}).X^2(z_o).b_{oo}\, dS$$

Table B.5 CONSTANT AGGREGATES

Shank Force

$$F_{qd}. = (F_v. + F_u. G_v.) N_b$$

$$F_{zqd}. = (F_{zv}. + F_u. G_{zv}.) N_b$$

$$F_q. = (F_b. - F_u. E_q^2.) N_b$$

$$F_{zq}. = (F_{zb}. + F_u. G_{zq}.) N_b$$

$$f_z = (f_{zo} + F_u. g_{bo}) N_b$$

$$f_o = (f_{io} + f_{sao}) N_b$$

Shank Moment

$$M_{qd}. = (M_v. + M_u. G_v.) N_b$$

$$M_{zqd}. = (M_{zv}. + M_u. G_{zv}.) N_b$$

$$M_q. = (M_b. - M_u. E_q^2.) N_b$$

$$M_{zq}. = (M_{zb}. + M_u. G_{zq}.) N_b$$

$$m_z = (m_{zo} + M_u. g_{bo}) N_b$$

$$m_o = (m_{io} + m_{ao}) N_b$$

Table B.6 AGGREGATE OPERATORS WHICH ARE FUNCTIONS OF RADIUS

$$F_p. = I. + F_u. M_g^{-1}. D_b^T.$$

$$M_p. = X(b_{oo}). + M_u. M_g^{-1}. D_b^T.$$

$$G_p. = M_g^{-1}. D^T.$$

Table B.7 STIFFNESS FUNCTIONS

$$K_{ed}. = K_e\ (f,q,S,N). - K_{ep}\ (S,N). \quad = \quad (K_{fed}^{T}.,K_{med}^{T}.)^{T}$$

$$G_k(f,q). = -M_g^{-1}. \int_o^R D^T(S). \int_o^R K_{ed}. D(N).\ dN\ dS$$

$$F_{sk}(f,q). = -N_b \int_o^R \int_o^R K_{fed}. D(N).\ dN\ dS + N_b F_u.\ G_k.$$

$$M_{sk}(f,q). = -N_b \int_o^R [X(b). \int_o^R K_{fed}. D(N).\ dN$$

$$+ \int_o^R K_{med}. D(N).\ dN] dS + N_b M_u . G_k.$$

Table B.8.1 BLADE NUMBER SELECT (Subsystem 1.1 of Figure B.3)

a) Produce the blade number count, i

$$i = 1,2 \ . \ . \ . \ . \ N_b$$

b) Produce the trig functions (sine and cosine) of Y_{bi},

$$\sin Y_{bi} = \sin Y_b \cos Y_{bj} + \cos Y_b \sin Y_{bj}$$

$$\cos Y_{bi} = \cos Y_b \cos Y_{bj} - \sin Y_b \sin Y_{bj}$$

$j = i - 1$ Constants: $\sin Y_b$, $\cos Y_b$

c) Load individual blade modal coordinates $q(N_m)$, $q^{\cdot}(N_m)$, $q^{\cdot\cdot}(N_m)$ into staging area $q_i(N_m)$, $q_i^{\cdot}(N_m)$, $q_i^{\cdot\cdot}(N_m)$

Table B.8.2 HUB TO SHANK TRANSFORMATIONS (Subsystem 1.2 of Figure B.3)

Inertial Velocity	Operation
$v_t = R_3\ (Y_t) \cdot v_h$	HAX TO TAX
$w_t - z = R_3\ (Y_t) \cdot w_h$	
$v_{tu} = v_t - X_3\ (Z_u) \cdot (w_t - z)$	TAX TO TUAX
$w_{tu} - z = w_t - z$	
$v_{cu} = R_2\ (Q_g) \cdot v_{tu}$	TUAX TO CUAX
$w_{cu} - z = R_2\ (Q_g) \cdot (w_{tu} - z) + I_2 \cdot \dot{Q}_g + R_2\ (Q_g)\ z - z$	
$v_c = v_{cu} + X_3\ (Z_u) \cdot (w_{cu} - z)$	CUAX TO CAX
$w_c - z = w_{cu} - z$	
$v_g = R_3\ (-Y_c) \cdot v_c$	CAX TO GAX
$w_g - z = R_3\ (-Y_c) \cdot (w_c - z)$	
$v_{si} = R_3\ (-Y_{bi}) \cdot v_g$	GAX TO SAX
$w_{si} - z = R_3\ (-Y_{bi}) \cdot (w_g - z)$	

Rotor Speed Variables:

$z = (0,\ 0,\ -Z)^T$; $\quad z^{\cdot} = (0,\ 0,\ -Z_d)^T$; $\quad Z = Z_o * Z_p + Z_o$

Geometry:

$Y_t = Y_c - Y$; $\quad Y_i = Y + Y_{bi}$; $\quad Y_{cbi} = Y_c + Y_{bi}$

Control:

$F_i = C_1 - C_2 \cos Y_i - C_3 \sin Y_i$

Table B.8.2 (continued)

Aerodynamic Velocity	Operation
$v_{at} = R_3\ (Y_t).\ v_{ah}$	HAX TO TAX
$w_{at} = R_3\ (Y_t).\ w_{ah} + z$	
$v_{atu} = v_{at} - X_3\ (Z_u).\ w_{at}$	TAX TO TUAX
$w_{atu} = w_{at}$	
$v_{acu} = R_2\ (Q_g).\ v_{atu}$	TUAX TO CUAX
$w_{acu} = R_2\ (Q_g).\ w_{atu} + I_2.\dot{Q}_g$	
$v_{ac} = v_{acu} + X_3\ (Z_u).\ w_{acu}$	CUAX TO CAX
$w_{ac} = w_{acu}$	
$v_{ag} = R_3\ (-Y_c).\ v_{ac}$	CAX TO GAX
$w_{ag}\ R_3\ (-Y_c).\ w_{ac}$	
$v_{asi} = R_3\ (-Y_{bi}).\ v_{ag}$	GAX TO SAX
$w_{asi} = R_3\ (-Y_{bi}).\ w_{ag}$	

Time Differentiation - Inertial Velocity	Operation
$\dot{v}_t = R_3\ (Y_t).\ (\dot{v}_h - X(z).\ v_h)$	HAX TO TAX
$\dot{w}_t = R_3\ (Y_t).\ (\dot{w}_h - X(z).\ w_h) + \dot{z}$	
$\dot{v}_{tu} = \dot{v}_t - X_3\ (Z_u).\ \dot{w}_t$	TAX TO TUAX
$\dot{w}_{tu} = \dot{w}_t$	
$\dot{v}_{cu} = R_2\ (Q_g).\ (\dot{v}_{tu} - X_2\ (\dot{Q}_g).\ v_{tu})$	TUAX TO CUAX
$\dot{w}_{cu} = R_2\ (Q_g).\ (\dot{w}_{tu} - X_2\ (\dot{Q}_g).\ w_{tu}) + I_2.\ddot{Q}_g$	
$\dot{v}_c = \dot{v}_{cu} + X_3\ (Z_u).\ \dot{w}_{cu}$	CUAX TO CAX
$\dot{w}_c = \dot{w}_{cu}$	
$\dot{v}_g = R_3\ (-Y_c).\ \dot{v}_c$	CAX TO GAX
$\dot{w}_g = R_3\ (-Y_c).\ \dot{w}_c$	
$\dot{v}_{si} = R_3\ (-Y_{bi}).\ \dot{v}_g$	GAX TO SAX
$\dot{w}_{si} = R_3\ (-Y_{bi}).\ \dot{w}_g$	

Table B.8.3 RADIAL STATION COUNTER (Subsystem 1.3 of Figure B.3)

Produce the radial position count: $j = 1,2, \ldots , N_r$

Table B.8.4 RADIAL FUNCTION GENERATOR (Subsystem 1.4 of Figure B.3)

Move all radially dependent functions from tables to a staging area for each j. The functions are,

Function	Size
$R_{oo}\cdot$	9
R_{fs}	1
$D_a\cdot$	3 * N_m
$D_b\cdot$	3 * N_m
b_{oo}	3
$G_p\cdot$	4 * N_m
$F_p\cdot$	9
$M_p\cdot$	9
f_{ao}	3
M_b	1
C_b	1

Table B.8.5 BLADE ELEMENT GEOMETRY (Subsystem 1.5 of Figure B.3)

$e_a = D_{a.}\ q^{\cdot}$	(B9.3)
$e_a^{\cdot} = D_{a.}\ q^{\cdot}$	
$e_a^{\cdot\cdot} = D_{a.}\ q^{\cdot\cdot}$	
$e_b = D_{b.}\ q$	(B9.2)
$e_b^{\cdot} = D_{b.} q^{\cdot}$	
$e_b^{\cdot\cdot} = D_{b.}\ q^{\cdot\cdot}$	
$R_o. = R_F\ (F - F_o).\ R_{oo}.$	(B3.1)
$R_e. = R_e\ (e_a)$	(DEFINITION)
$R. = R_e.\ R_o.$	(B3.2)
$R_F. = R_1\ (F_s).$	(MODEL DEFINITION)
$R_p. = R_e. - I. + X(e_a).$	(DEFINITION)
$b_p = x + e_b$	(B13.14, 15.1)
$b^{\cdot} = x^{\cdot} + e_b^{\cdot}$	(B15.2)
$b = b_p + b_{oo}$	(B13.14)
$w_{bs} = W_y(e_a).\ e_a^{\cdot}$	(B9.11)
$w_{bs}^{\cdot} = W_y(e_a).\ e_a^{\cdot\cdot}$	(B9.12)
$x' = R_{oo}^T.\ (-R_e^T.\ (R_F^T. - I.)\ -R_p^T.\ )i_1$	(B15.1,9.1,9.8)
$X^{\cdot\prime} = (-R.^T\ X(w_{bs}). + R_{oo}^T.X(e_a^{\cdot}).\ )i_1$	(B15.2,9.9,9.8)
$x^{\cdot\cdot\prime} = (-R.^T\ (X^2(w_{bs}^{\cdot}).\ )+ R_{oo}^T.\ X(e_a^{\cdot\cdot}).\ )i_1$	(B15.3,9.10,9.8)

Trig Model for Feathering Factor, F_s

$F_s = (F-F_o)\ R_{fs}\ (S)$	(DEFINITION)
$F_{sq} = F_s^2$	(DEFINITION)
$SF = \sin F_s = F_s * (1 - F_{sq}(1-F_{sq}/20)/6)$	(MODEL APPROX.)
$CF = \cos F_s = 1 - F_{sq}(1-F_{sq}/12)/2$	(MODEL APPROX)

Table B.8.6 SHANK TO BLADE TRANSFORMATIONS (Subsystem 1.6 of Figure B.3)

$v_a = R.\ (v_{as} + \dot{b} + X(w_{as}).\ b\ ) + v_{aa}$ (B7.1,2)

$u_s = \dot{v}_s + X(w_s - z).\ v_s + X(z).\ v_s$ (B6.4)

$w_s - z = w_{si} - z$ (DEFINITION)

$\dot{w}_s = \dot{w}_{si}$ (DEFINITION)

$v_{as} = v_{asi}$ (DEFINITION)

$w_{as} = w_{asi}$ (DEFINITION)

Table B.8.7 DISTRIBUTED INERTIAL LOADS (Subsystem 1.7 of Figure B.3)

$$u_{mbr} = u_s + \ddot{x} + 2X(w_s - z).\ \dot{b} + 2X(z).\ \dot{x} + X(\dot{w}_s).\ b$$
$$+ (\ X(w_s - z).\ X(w_s).\ + X(z).\ X(w_s - z).\ )b + X^2(z).\ x$$

(B15.4)

$f_{mi} = -M_b\ u_{mbr}$ (B12.2)

$p_{mi} = (f_{mi}^T,\ m_i^T)^T$ (B6.5)

$m_i = 0$ (ASSUMPTION)

Table B.8.8 AERO GEOMETRY (Subsystem 1.8 of Figure B.3)

$$M_2 = V_{a2} R_{vc} \qquad \text{(DEFINITION)}$$

$$M_3 = V_{a3} R_{vc} \qquad \text{(DEFINITION)}$$

$$M^2 = M_2^2 + M_3^2 \qquad \text{(B7.3)}$$

$$\sin A_b = M_3 / M \qquad \text{(B7.4)}$$

$$\cos A_b = M_2 / M \qquad \text{(B7.4)}$$

Table B.8.9 AERODYNAMIC COEFFICIENT TABLES (Subsystem 1.9 of Figure B.3)

Logical operations on sin A_b and cos A_b (select one) for table look-up operand. Pointers to proper tables are also established by this logic. The selected operand O_A (either sin A_b or cos A_b) is used with M for three bivariate look-ups.

$$C_{fa2} = C_{fa2} (O_A, M) \qquad \text{(B7.7)}$$

$$C_{fa3} = C_{fa3} (O_A, M)$$

$$C_{ma1} = C_{ma1} (O_A, M) \qquad \text{(B7.8)}$$

$$C_{fa1} = C_{ma2} = C_{ma3} = 0$$

Table B.8.10 DISTRIBUTED AERO LOADS (Subsystem 1.10 of Figure B.3)

$$Q_{ac} = 1481 M^2 P_r C_b \qquad \text{(B7.5,6)}$$

$$f_{ma} = Q_{ac} R^T \cdot R_1^T(A_{b1}) \cdot C_{fa} - f_{ao} \qquad \text{(B7.9,12.3)}$$

$$m_a = Q_{ac} C_b c_{ma} \qquad \text{(B7.10)}$$

$$p_{ma} = (f_{ma}^T, m_a^T)^T \qquad \text{(B7.11)}^*$$

Table B.8.11 LOADS INTEGRAND ASSEMBLER (Subsystem 1.11 of Figure B.3)

$$g_r' = G_p. (p_{mi} + p_{ma}) \qquad (B13.6)^*$$

$$f_r' = F_p. f_{mb} \qquad (B13.8)^*$$

$$\begin{aligned} m_r' = {} & (M_p. + X(b_p).)\ f_{mb} \\ & + X(b_p).(-M_b(e_b^{\cdot\cdot} + X(z).(2\ e_b^{\cdot} + X(z).\ e_b))) \\ & + X(x).(-M_b\ X^2(z).\ b_{oo} + f_{ao}) \end{aligned} \qquad (B13.13,1,2)$$

$$f_{mb} = f_{mi} + f_{ma} \qquad {}^*(DEFINITION)$$

Table B.8.12 RADIAL INTEGRALS (Subsystem 1.12 of Figure B.3)

$$x = \int_0^S x'\ dW$$

$$x^{\cdot} = \int_0^S x^{\cdot\prime}\ dW \qquad \text{Geometry} \qquad \text{(Definitions)}$$

$$x^{\cdot\cdot} = \int_0^S x^{\cdot\cdot\prime}\ dW$$

$$g_{ri} = \int_o^R g_r'\ dS \qquad \text{Modal} \qquad \text{(Definition)}$$

$$f_{ri} = \int_o^R f_r'\ dS$$

$$\text{Gimbal Loads} \qquad \text{(Definitions)}$$

$$m_{ri} = \int_o^R m_r'\ dS$$

Table B.8.13 MODAL COORDINATE INTEGRATIONS (Subsystem 1.13 of Figure B.3)

Solve for every blade, i, $i = 1,2,..N_b$

$$g_i = g_{ri} + (Z_p\ G_{zv}. + G_v.)\ \dot{q}_i + Z_z(g_{bo} + G_{zq}.\ q_i)$$

$q_{si} = q_i$ (Save q_i)

$$q_i = K_{qq}.\ q_{si} + (K_{qv}.+Z_p\ K_{qzv}.)\dot{q}_i + (K_{qg}.+Z_p\ K_{qzg}.)g_i$$

$$\dot{q}_i = K_{vv}.\ \dot{q}_i + (K_{vq}.+Z_p\ K_{vzq}.)q_{si} + (K_{vg}.+Z_p\ K_{vzg}.)g_i$$

$$\ddot{q}_i = -\ E_q^2.\ q_i + g_i$$

Expressions for the constants in Modal Coordinate Integrations:

Given: P_o, Z_o, D_t, K_{PZ}, N_{IS}, G_{qd}, G_q

Solve: $Y_d = Z_o\ D_t$ $S_a = \sin\ (A)$ $W = P_o\ Z_o$

$A = PY_d$ $C_a = \cos\ (A)$ $P = N_{IS}\ P_o$

Unconstrained (NIS > 0)

$K_{qq} = C_a$

$K_{qv} = S_a\ /\ W$

$K_{qg} = (1.-C_a)\ /\ W^2$

$K_{vq} = -W$

$K_{vg} = K_{qv}$

$P_{pzp} = (Z_o\ K_{PZ} + P)\ /\ W^2$

$K_{qzv} = Z_o\ P_{pzp}\ (A\ C_a - S_a)$

$K_{qzg} = Z_o\ P_{pzp}\ (AS_a+2C_a-2)/W$

$K_{vzq} = -Z_o\ P_{pzp}$

$K_{vzg} = K_{qzv}$

$E_q^2 = W^2$

$K_{vv} = C_a$

Constrained (NIS = 0)

$W_m^2 = W^2 + G_q$

$T_q = G_{qd}\ /\ W_m^2$

$K_{qq} = \exp(-Y_d/(T_q\ Z_o))$

$K_{qv} = 0$

$K_{qzv} = 0$

$K_{qg} = (1.-K_{qq})\ /\ W_m^2$

$K_{vv} = 0$

$K_{vq} = -1./T_q$

$K_{vg} = 1./(T_q\ W_m^2)$

$K_{vzq} = -2Z_o^2\ K_{PZ}\ /\ W_m^4\ G_{qd}$

$K_{vzg} = 0$

$K_{qzg} = 0$

Table B.8.14 MULTIBLADE MODAL COMBINATIONS (Subsystem 1.14 of Figure B.3)

$$q_s = (1/N_b)\ \sum_i q_i \sin Y_{bi} \qquad \text{(B14.3)}$$

$$q_c = (1/N_b)\ \sum_i q_i \cos Y_{bi} \qquad \text{(B14.4)}$$

$$q_o = (1/N_b)\ \sum_i q_i \qquad \text{(B14.5)}$$

$$q_{ds} = (1/N_b)\ \sum_i \dot{q}_i \sin Y_{bi} \qquad \text{(B14.6)}$$

$$q_{dc} = (1/N_b)\ \sum_i \dot{q}_i \cos Y_{bi} \qquad \text{(B14.7)}$$

$$q_{do} = (1/N_b)\ \sum_i \dot{q}_i \qquad \text{(B14.8)}$$

$$q_m = (q_o^T,\ q_c^T,\ q_s^T)^T \qquad \text{(B14.9)}$$

$$q_{md} = (q_{do}^T,\ q_{dc}^T,\ q_{ds}^T)^T \qquad \text{(B14.10)}$$

Table B.8.15 MULTIBLADE LOADS COMBINATIONS (Subsystem 1.15 of Figure B.3)

$$f_{gr} = \sum_i R_3(Y_{bi})\ f_{ri} \qquad \text{(B14.12)}$$

$$m_{gr} = \sum_i R_3(Y_{bi})\ m_{ri} \qquad \text{(B14.13)}$$

Table B.8.16 GIMBAL LOADS SUMMATION (Subsystem 1.16 of Figure B.3)

$$f_g = f_{gr} + A_m(F_{sq}.).\ q_m + A_m(F_{sqd}.).\ q_{md} + I_3.\ f_{so} \qquad \text{(B14.12)}^*$$

$$m_g = m_{gr} + A_m(M_{sq}.).\ q_m + A_m(M_{sqd}.).\ q_{md} + I_3.\ m_{so} \qquad \text{(B14.13)}$$

where,

$$A_m\ (A.).\ = \begin{bmatrix} 0 & A_1 & A_2 \\ 0 & A_2 & -A_1 \\ A_3 & 0 & 0 \end{bmatrix}$$

$$F_{sq}. = F_q. + F_{zq}.\ Z_z \qquad \text{(B13.10)}$$

$$F_{sqd}. = F_{qd}. + F_{zqd}.\ Z_p \qquad \text{(B13.9)}$$

$$f_{so} = f_z\ Z_z + f_o \qquad \text{(B13.11)}$$

$$M_{sq}. = M_q. + M_{zq}.\ Z_z \qquad \text{(B13.15)}$$

$$M_{sqd}. = M_{qd}. + M_{zqd}.\ Z_p \qquad \text{(B13.16)}$$

$$m_{so} = m_z\ Z_z + m_o \qquad \text{(B13.17)}$$

Table B.8.17 GIMBAL LOADS TRANSFORMATIONS (Subsystem 1.17 of Figure B.3)

$$f_c = R_3\ (Y_c) \cdot f_g$$

$$m_c = R_3\ (Y_c) \cdot m_g$$

$$f_{cu} = f_c$$

$$m_{cu} = m_c + X_3(Z_u) \cdot f_c$$

$$f_{tu} = R_2\ (-Q_g) \cdot f_{cu}$$

$$m_{tu} = R_2\ (-Q_g) \cdot m_{cu}$$

$$f_t = f_{tu}$$

$$m_t = m_{tu} + X_3(-Z_u) \cdot f_{tu}$$

$$f_h = R_3\ (-Y_t) \cdot f_t$$

$$m_h = R_3\ (-Y_t) \cdot m_t$$

$$E_G = i_2^T\ m_{cu}$$

$$Y_t = Y_c - Y$$

Table B.8.18 GIMBAL SERVO INTEGRATOR (Subsystem 1.18 of Figure B.3)

$i = NM + 1, \qquad j = 2 * NM + 1$

$$\ddot{Q}_g = K_{ge}\ E_g + K_{gqc}\ Q_{mi} + K_{ggs}\ Q_{mj} + K_{gvc}\ Q_{mdi} + K_{gvs}\ Q_{mdj}$$

$$Q_{sg} = Q_g \qquad \text{(Save } Q_g\text{)}$$

$$Q_g = G_{qq}\ Q_{sg} + (G_{qv} + Z_p\ G_{qzv})\dot{Q}_g + (G_{qa} + Z_p\ G_{qza})\ddot{Q}_g$$

$$\dot{Q}_g = G_{vv}\ \dot{Q}_g + (G_{vq} + Z_p\ G_{vzq})Q_{sg} + (G_{va} + Z_p\ G_{vza})\ddot{Q}_g$$

APPENDIX C

GIMBAL KINEMATICS ANALYSIS

Figures

Tables

Presented in this Appendix is an analysis of the "gimbal kinematics", and the the manner in which the blade root loads contribute to a "gimbal error". The kinematics equations depend specifically on the detailed design of the gimbal which, of course, can take many arrangements.

Two relatively common systems are analyzed below. These are the teetering (with "flap-pitch coupling" and "undersling") and the focused floating gimbal.

C.1. The Teetering System

Figure C.1 depicts the classical teetering rotor system with flap-pitch coupling (Y_c) and undersling Z_u. The figure also presents some of the various axis systems used in the kinematic analysis. "Rotor" axes (r) are shown in the conventional position. Rotor axes are fixed to the shaft, with r_3 coincident with the shaft centerline.

"Teetering" axes (t) are also fixed to the shaft, but have been rotated by the angles Y_c about t_3, so that t_2 is parallel to the teetering hinge.

"C" axes are fixed to the gimbal outer housing, as are the rotor blades. When there is no gimbal deflection (i.e., $Q_g = 0$), C axes are coincident with teetering axes, but as the gimbal moves, the C axes rotate with respect to teetering axes, by the angle Q_g. Note that, if there is undersling, then C axes also translate with respect to teetering axes.

Figure C.2, used later in the analysis, presents a clearer view of the relationship between teetering and C axes.

"Gimbal axes" (sub g) are fixed to the outer gimbal ring (blade-root mounting ring), but are rotated the angle $-Y_c$ from C axes about g_3. Hence gimbal axes appear to the rotor blade analysis, exactly as rotor axes would with no gimbal present.

To help further clarify the definitions of these various coordinate systems, Table C.1 summarizes the systems and presents abbreviations and subscripts used throughout the subsequent kimematics analysis.

"Shank axes" (s) associate with each blade shank - the region of the blade connected to the outer gimbal housing. For blade number i, $i=1,2,..N_b$, the shank axes are rotated Y_{bi} about g_3 from gimbal axes. The projection of the blade reference line on the 1-2 plane generally lies along the negative s_1 axis.

C.1.1 Motion Transformations

The reader is directed to Appendix A, which presents general procedures for entity definitions and transformations. These standardized methods are convenient for derivation of equations in a form suitable for direct computer programming.

Figures C.1 and C.2 have been used to derive the motion transformation expressions of Table C.2 in accordance with the procedures of Appendix A.

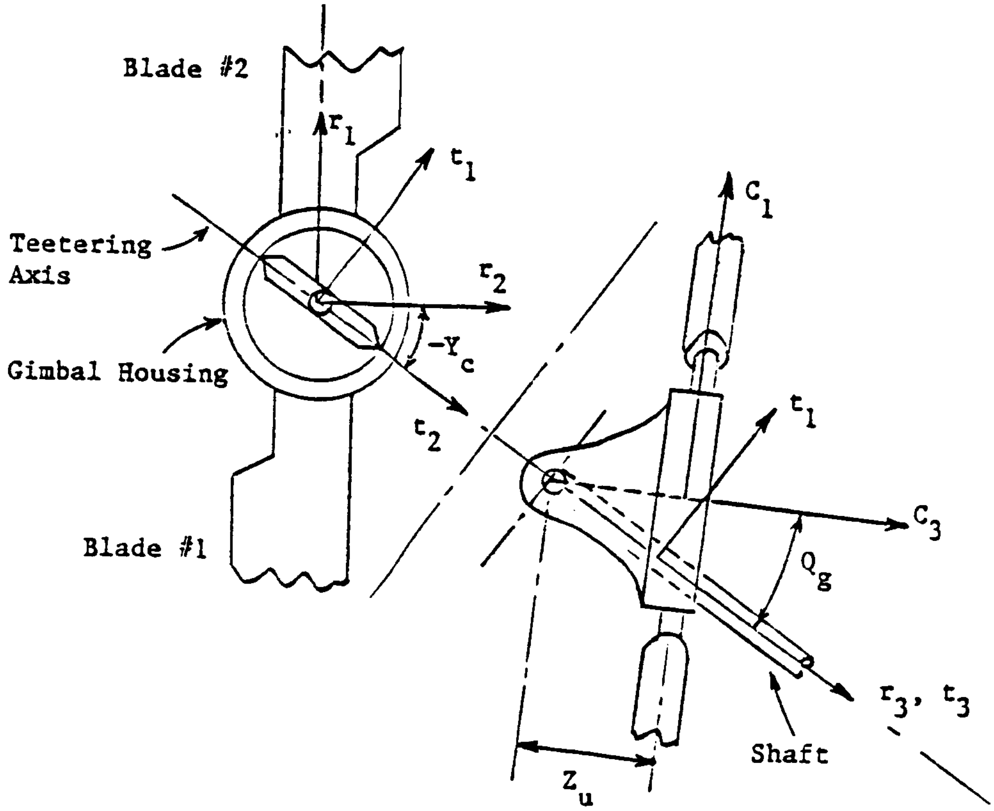


Figure C.1. Conventional Teetering Rotor Hub

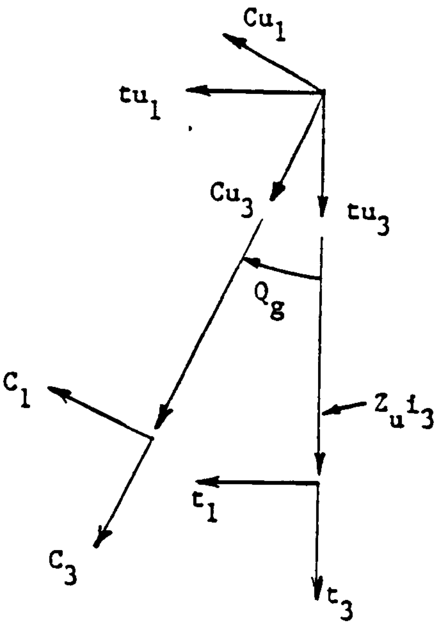


Figure C.2. Coordinates and Notations for Motion Summation

C.1.2 Loads Transformations - Gimbal Errors

Loads produced by all blades in the rotor are applied to the outer gimbal ring at the blade shanks. These loads sum to resultant force and moment columns resolved to gimbal axes. The loads transformation equations convert the gimbal loads resolved to hub axes, representing the forces and moments the rotor applies to its supports. Table C.3 presents the loads transformations. Note that a "gimbal error" load is also provided.

The presence of a gimbal in the system essentially introduces into the dynamics equations a requirement that certain moments acquire specific attributes. For example, a simple free teetering system requires that the moment about the teetering hinge vanishes. If the hinge motion is constrained, the moment remains zero between stops and then acquires the attribute of a very stiff spring upon contacting a stop. Other attributes can also be demanded by the design of the gimbal. An example is a system with a spring installed about a gimbal axis.

The loads transformation expressions of Table C.3 produce the moment about the teetering axis as a gimbal error function: An external calculation (the servo) converts this error into a gimbal acceleration to produce the desired overall system behaviour.

C.2 The Floating Gimbal

As described previously, the detailed design characteristics of the gimbal system must usually be considered in the dynamic math model of such a system. The teetering gimbal was analyzed in Section C.1. The same basic expressions derived for the teetering system are easily expanded for use in analyzing a hypothetical "floating gimbal" with two degrees of freedom.

Figure C.3 shows the floating system with a "focal point" not at the center of the rotor hub. As such, this system is similar to the teetering system, except that it frees the moments in the ball joint in two directions and, hence, has two-element gimbal error and degree of freedom columns.

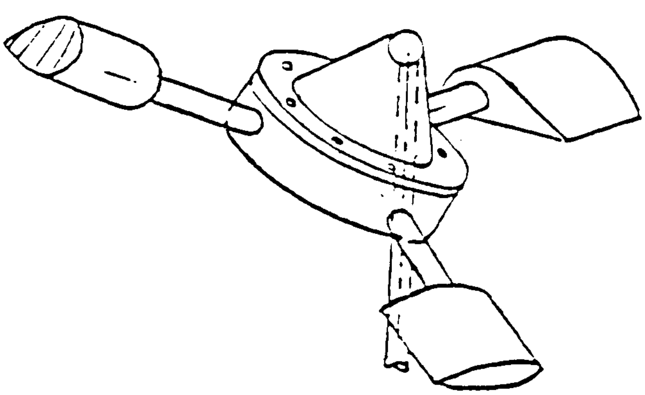


Figure C.3 - Sketch of Floating Gimbal System

The math model of the floating gimbal is very similar to that of the teetering system. The gimbal degree of freedom variables , Q_g, $\dot{Q}_g$, and $\ddot{Q}_g$ now become vector columns with two non-zero elements, representing motions about both the t_1 and t_2 axes. The transformation array R_2, must be expanded to consider an 1-axis rotation also. The new transformation matrix R_{12}, can be used in place of R_2 in the previous equations. If the first element in the q_g column, Q_{g1}, represents the 1-rotation and Q_{g2} the 2-rotation, the new transformation array is written:

$$R_{12}. = R_1(Q_{g1}). \quad R_2(Q_{g2}).$$

With these definitions, the teetering matrix equations can be applied directly to the floating system.

The gimbal loads expressions can also be applied; in this case the two-element gimbal error column is composed of both 1 and 2- axis components of the moment at the ball joint.

Table C.1 COORDINATE SYSTEMS USED IN GIMBAL SYSTEM ANALYSIS

No.	Name	Abbreviation	Axis Label	Fixed To
1	Hub Axes	HAX	h	Nonrotating Frame
2	Rotor Axes	RAX	r	Rotating Shaft
3	Gimbal Axes	GAX	g	Outer Gimbal Ring
4	Shank Axes	SAX	s	Outer Gimbal Ring (s_3 coincident with g_3)
5	Teetering Axes	TAX	t	Rotating Shaft
6	C Axes	CAX	c	Outer Gimbal Ring

The following axes systems account for the rotor undersling.

7	TU Axes	TUAX	tu	
8	CU Axes	CUAX	cu	

HAX ... Origin and h_3 coincident with corresponding RAX entities, but HAX do not spin with the rotor. HAX are rotated the angle Y from RAX about h_3, where Y = Z T and Z is the rotor spin rate.

RAX ... Origin at hub center when gimbal is undeflected. Negative r_1 axis lies under blade number 1 projection on the shaft normal plane.

GAX ... Origin and g_3 axis coincident with corresponding CAX entities. Rotated an angle Y_c from CAX about the C_3 g_3 axis. As defined, GAX are coincident with RAX when gimbal is undeflected.

SAX ... GAX rotated Y_{bi} about g_3 for each blade, i, to become shank axes for blade i. Shank axes for blade number 1 are GAX.

TAX ... Origin and t_3 coincident with r_3 and RAX origin. Rotated the angle Y_c (about t_3) from RAX so that t_2 is parallel to the teetering hinge.

CAX ... Coincident with TAX when gimbal is undeflected.

Table C.2 HUB TO SHANK MOTION TRANSFORMATIONS

Inertial Velocity	Operation
$v_t = R_3\ (Y_t).\ v_h$	HAX TO TAX
$w_t = R_3\ (Y_t).\ w_h + z$	
$v_{tu} = v_t - X_3\ (Z_u).\ w_t$	TAX TO TUAX
$w_{tu} = w_t$	
$v_{cu} = R_2\ (Q_g).\ v_{tu}$	TUAX TO CUAX
$w_{cu} = R_2\ (Q_g).\ w_{tu} + I_2.\dot{Q}_g$	
$v_c = v_{cu} + X_3\ (Z_u).\ (w_{cu}-z)$	CUAX TO CAX
$w_c = w_{cu}$	
$v_g = R_3\ (-Y_c).\ v_c$	CAX TO GAX
$w_g = R_3\ (-Y_c).\ w_c$	
$v_{si} = R_3\ (-Y_{bi}).\ v_g$	GAX TO SAX
$w_{si} = R_3\ (-Y_{bi}).\ w_g$	

Rotor Speed Variables:

$z = (0,\ 0,\ -Z)^T$; $z^{\bullet} = (0,\ 0,\ -Z_d)^T$; $Z = Z_o * Z_p + Z_o$

Geometry:

$Y_t = Y_c - Y$; $Y_i = Y + Y_{bi}$; $Y_{cbi} = Y_c + Y_{bi}$

Control:

$F_i = C_1 - C_2 \cos Y_i - C_3 \sin Y_i$

Table C.2 (continued)

Aerodynamic Velocity	Operation
$v_{at} = R_3\ (Y_t) \cdot v_{ah}$	HAX TO TAX
$w_{at} = R_3\ (Y_t) \cdot w_{ah} + z$	
$v_{atu} = v_{at} - X_3\ (Z_u) \cdot w_{at}$	TAX TO TUAX
$w_{atu} = w_{at}$	
$v_{acu} = R_2\ (Q_g) \cdot v_{atu}$	TUAX TO CUAX
$w_{acu} = R_2\ (Q_g) \cdot w_{atu} + I_2 \cdot \dot{Q}_g$	
$v_{ac} = v_{acu} + X_3\ (Z_u) \cdot w_{acu}$	CUAX TO CAX
$w_{ac} = w_{acu}$	
$v_{ag} = R_3\ (-Y_c) \cdot v_{ac}$	CAX TO GAX
$w_{ag}\ R_3\ (-Y_c) \cdot w_{ac}$	
$v_{asi} = R_3\ (-Y_{bi}) \cdot v_{ag}$	GAX TO SAX
$w_{asi} = R_3\ (-Y_{bi}) \cdot w_{ag}$	

Time Differentiation - Inertial Velocity	Operation
$\dot{v}_t = R_3\ (Y_t) \cdot (\dot{v}_h - X(z) \cdot v_h)$	HAX TO TAX
$\dot{w}_t = R_3\ (Y_t) \cdot (\dot{w}_h - X(z) \cdot w_h) + \dot{z}$	
$\dot{v}_{tu} = \dot{v}_t - X_3\ (Z_u) \cdot \dot{w}_t$	TAX TO TUAX
$\dot{w}_{tu} = \dot{w}_t$	
$\dot{v}_{cu} = R_2\ (Q_g) \cdot (\dot{v}_{tu} - X_2\ (\dot{Q}_g) \cdot v_{tu})$	TUAX TO CUAX
$\dot{w}_{cu} = R_2\ (Q_g) \cdot (\dot{w}_{tu} - X_2\ (\dot{Q}_g) \cdot w_{tu}) + I_2 \cdot \ddot{Q}_g$	
$\dot{v}_c = \dot{v}_{cu} + X_3\ (Z_u) \cdot \dot{w}_{cu}$	CUAX TO CAX
$\dot{w}_c = \dot{w}_{cu}$	
$\dot{v}_g = R_3\ (-Y_c) \cdot \dot{v}_c$	CAX TO GAX
$\dot{w}_g = R_3\ (-Y_c) \cdot \dot{w}_c$	
$\dot{v}_{si} = R_3\ (-Y_{bi}) \cdot \dot{v}_g$	GAX TO SAX
$\dot{w}_{si} = R_3\ (-Y_{bi}) \cdot \dot{w}_g$	

Table C.3 GIMBAL LOADS TRANSFORMATIONS

$$f_c = R_3 (Y_c) \cdot f_g$$

$$m_c = R_3 (Y_c) \cdot m_g$$

$$f_{cu} = f_c$$

$$m_{cu} = m_c + X_3(Z_u) \cdot f_c$$

$$f_{tu} = R_2 (-Q_g) \cdot f_{cu}$$

$$m_{tu} = R_2 (-Q_g) \cdot m_{cu}$$

$$f_t = f_{tu}$$

$$m_t = m_{tu} + X_3(-Z_u) \cdot f_{tu}$$

$$f_h = R_3 (-Y_t) \cdot f_t$$

$$m_h = R_3 (-Y_t) \cdot m_t$$

$$E_G = i_2^T m_{cu}$$

$$Y_t = Y_c - Y$$

APPENDIX D

GIMBAL SUBSYSTEM DYNAMICS

The Appendix contains the "Test and Verification Report: A Gimbal Subsystem for the Wind Energy System Time-Domain (WEST) Analyzer" by John A. Hoffman (PPI-1640-3; December 1979). Analysis included herein form the bases for the gimbal servo algorithms presented elsewhere in this report for use in solving the aeroelastic rotor math models including gimballed articulation.

1.0 INTRODUCTION

This report documents the experimental implementation of a rotor gimbal subsystem into the Wind Energy System Time-Domain (WEST) analyzer. The WEST 2 unit was used in the implementation.

References 1 and 2 document the math models and hybrid computer.implementation diagrams for the WEST 1 and WEST 2 simulator units. The original WEST math models embraced articulated and hingeless rotor blade retention systems; gimballed rotors (teetering or floating) were not addressed by the original simulators.

Subsequent to the design of the initial WEST concepts, teetering emerged as a viable design alternative in U.S. wind energy systems. Thus, an additional subsystem was designed for the WEST units that would allow them to analyze teetering rotors. Reference 2 presents the math model derivations and hybrid program diagrams for the teetering subsystem.

The teetering subsystem for the full analysis was simplified to an "abbreviated" system, so that it would fit on a single General Purpose Electronic System (GPURS). The GPURS implementation had the purpose of proving the concept of the gimbal subsystem, prior to its being hardwired in its complete form within the WEST units themselves.

This report deals with this abbreviated GPURS implementation, which is valid if the rotor shaft remains fixed in space.* Although abbreviated, the system proves the feasibility of the concept so that the full system can be hardwired with confidence, without further feasibility testing.

Although considerable difficulty was encountered with the dynamic stability of the gimbal/WEST simulation, techniques were developed which enabled the system to perform very well.

*The full subsystem allows shaft motions.

The remainder of this report is organized as follows:

Section 2.0 presents a description of the basic gimbal subsystem, drawing on the detailed developments in Reference 2.

Section 3.0 presents performance results, in the form of oscilloscope photographs, which show correct operation of the gimbal subsystem in a closed-loop simulation with WEST 2.

In Section 4.0, the technical problems encountered in implementing the gimbal subsystem and their solutions are presented.

Section 5.0 presents conclusions and recommendations for the future of the gimbal subsystem in the WEST units.

2.0 DESCRIPTION OF THE GIMBAL SUBSYSTEM FOR WEST

Reference 2 presents the complete derivation of the equations for the WEST gimbal subsystem and hybrid computer implementation diagrams for the subsystem. The gimbal subsystem is essentially a "drop-in" package between the existing rotor analysis and the shaft support system math models. Configured in this way, only minimal modification of existing hybrid systems (for the wind turbine and shaft supports) is required.

Figure 2.1 presents the fundamental technique incorporated in the gimbal subsystem. The existing rotor analysis calculates the loads that blades place on the rotor hub. Proper resolution of these produces the total moment about the teetering hinge, which should be zero because of the hinge. This moment is called the "gimbal error," ϵ.

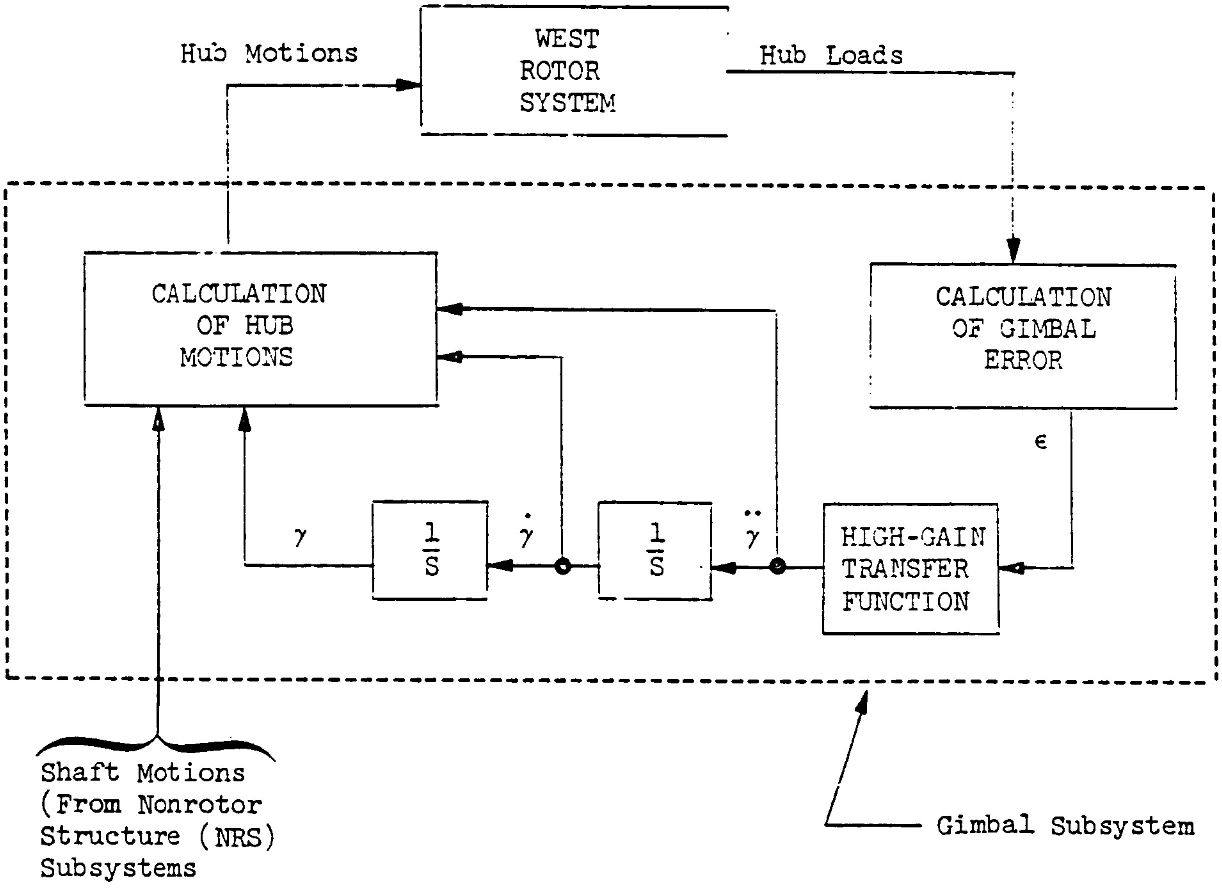

Hub Motions
WEST ROTOR SYSTEM
Hub Loads
CALCULATION OF HUB MOTIONS
CALCULATION OF GIMBAL ERROR
ϵ
HIGH-GAIN TRANSFER FUNCTION
$\frac{1}{S}$
$\frac{1}{S}$
γ
$\dot{\gamma}$
$\ddot{\gamma}$
Shaft Motions (From Nonrotor Structure (NRS) Subsystems
Gimbal Subsystem

Figure 2.1 - Basic Configuration of WEST Gimbal Subsystem

The gimbal acceleration, $\ddot{\gamma}$, is calculated by multiplying the error by an arbitrarily large gain so that for practical $\ddot{\gamma}$ values, $\epsilon \rightarrow 0$ as the gain becomes higher. $\dot{\gamma}$ and γ are produced by integrating $\ddot{\gamma}$, and these three gimbal motion entities are combined with motions of the shaft to produce rotor hub motions. Rotor hub motions feed back to the rotor analysis, closing the loop.

Figure 2.2 presents a simple model of the rotor blades, assumed perfectly rigid for the example. Figure 2.3 shows a more quantitative version of Figure 2.1, using the simple rigid rotor representation. As is easily perceived from Figure 2.3, $\epsilon \rightarrow 0$ as $\lambda \rightarrow \infty$, so that the system solves the equation

$$f = I\ddot{\gamma} + d\dot{\gamma} + k\gamma$$

as the loop gain, λ, is increased.

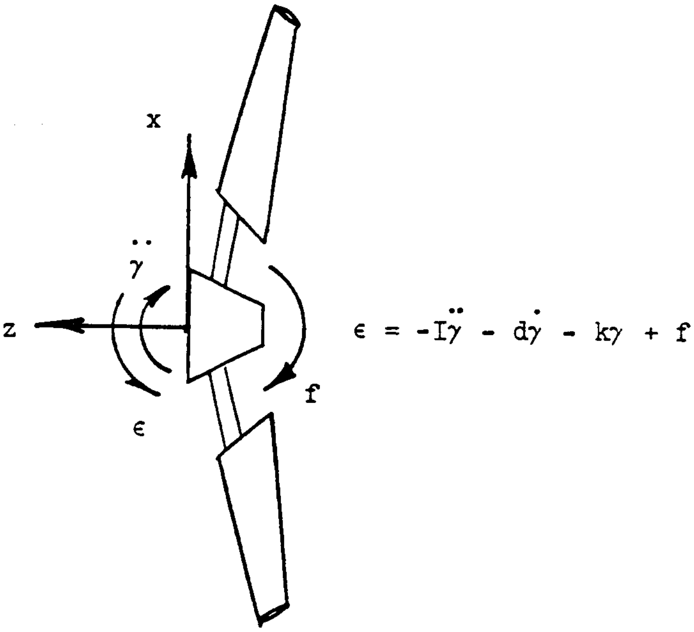


Figure 2.2 - Simple Quantitative Model
For Windturbine Rotor

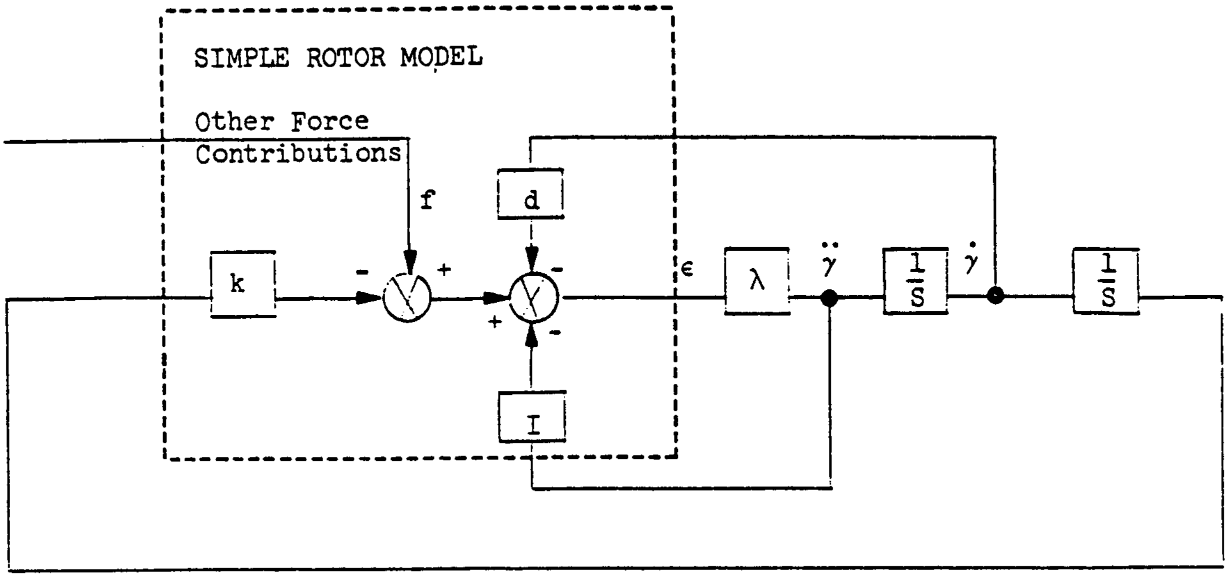


Figure 2.3 - Gimbal Analysis With Simplified Rigid Rotor Model

The final implementation diagrams for the abbreviated gimbal system are presented in Appendix A. These diagrams embrace the concept of Figure 2.1, using the detailed transformation expressions of Reference 2.

3.0 PERFORMANCE OF THE ABBREVIATED GIMBAL SUBSYSTEM

After some modification (described in Section 4.0), but with the basic concept intact, the abbreviated gimbal subsystem was found to perform well with the WEST 2 rotor and NRS systems. Figure Series 3 presents oscilloscope photographs which document the performance of the full coupled system. These results are discussed below.

"Nominal conditions" have been used in Reference 3 to correlate WEST with other analyses. These conditions include a windspeed of 27 mph, 40 rpm rotor speed, 100 percent shadow retardation through an 18-degree sector, and 20 percent windshear profile. Normalizing parameters on the traces are

R = Rotor Radius = 62.5 ft

W = Normalizing Thrust = 5,000 lb

Figures 3.1 and 3.2 show nominal operating conditions with gimbal on and off. Note the reduction of the gimbal error of Figure 3.2 (gimbal off) when compared to the trace of Figure 3.1. The high frequency "serpentine"* mode is the predominant error function. Figure 3.1 reveals that this lightly damped mode has a frequency of about 16 hz. This mode is excited twice per rev by the shadow effect, which produces an impulsive forcing function on the system. The serpentine ringing does not appear if the shadow effect is off, and only windshear (which is a pure 1P signal with no impulse) turned on.

Figures 3.3 and 3.4 reveal the performance of the gimbal system in reducing 1P gimbal errors. Figure 3.3 shows a case with extreme windshear, producing a 1P gimbal error of about 7v peak when the gimbal system is off.

*The serpentine mode is the high frequency mode where the gimbal moves in one direction, and the blade flapwise bending modes move in opposite directions.

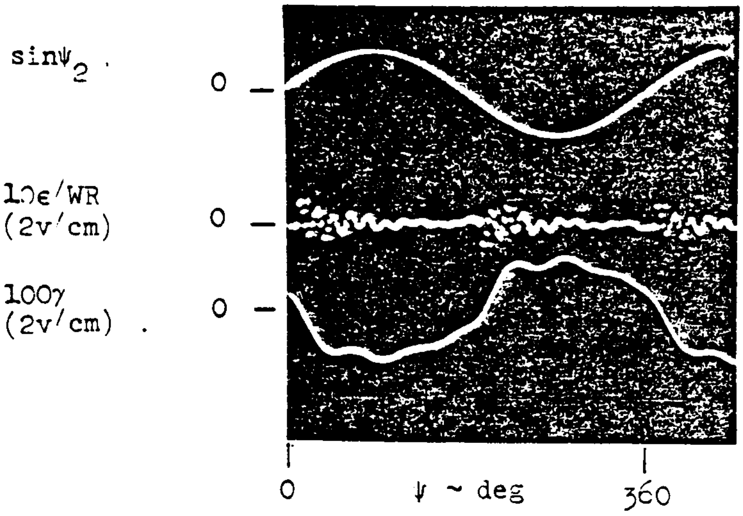


Figure 3.1 - Nominal Case - Gimbal Subsystem On

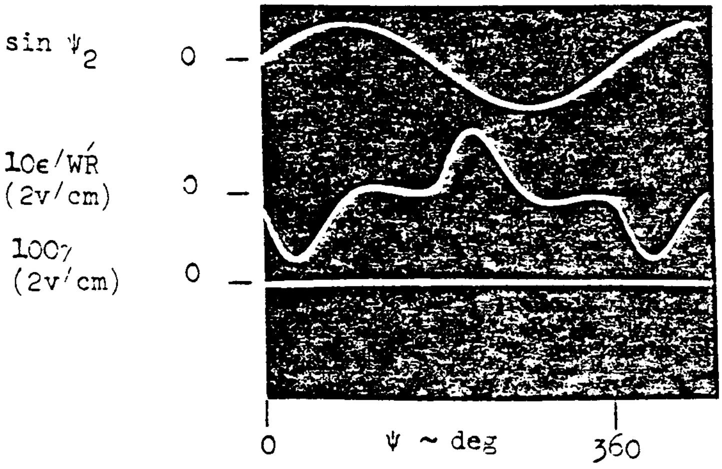


Figure 3.2 - Nominal Case - Gimbal Subsystem Off

Figures 3.1 and 3.2	REP. NO.
GIMBAL PERFORMANCE EXAMINATIONS - NOMINAL CONDITIONS	

ORIGINAL PAGE IS
OF POOR QUALITY

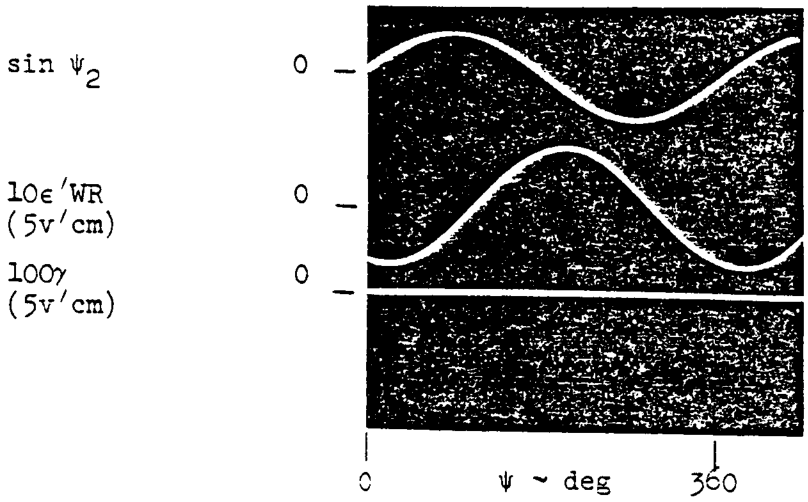


Figure 3.3 - No Shadow, No Gimbal, Extreme Windshear Case

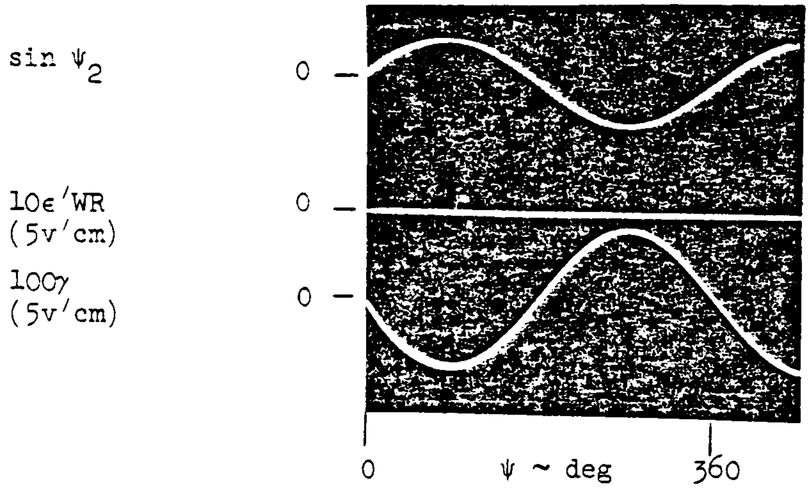


Figure 3.4 - Same As Figure 3.3 Except Gimbal On

Figures 3.3 and 3.4	REP. NO.
GIMBAL PERFORMANCE EXAMINATIONS - 1 PER REV EXCITATION	

With the gimbal system turned on, the error is reduced substantially, as shown by Figure 3.4. A smaller scale on error revealed that the 1P gimbal error is reduced to about 50 mv peak. Hence, the closed-loop gimbal subsystem reduces gimbal errors by over a factor of 1000, or 60 db.

Figures 3.5 and 3.6 reveal the benefits of the gimbal in reducing blade root flap bending moments. The peak to peak moment with the gimbal is about 3v. Without the gimbal, the moment is about 3.8v, so the gimbal reduces the moment by about 27 percent.

Studies of Reference 4 indicated a potential benefit of reducing blade root out-of-plane bending moments by 1/2, using gimballing. This benefit is realized only for bending normal to the plane of rotation. The moments in Figures 3.5 and 3.6 are measured parallel to the blade chord, which is situated at an angle of about 30 degrees with respect to the normal plane. Hence, half of the 1P gravity moment is supported as a 1P flapwise bending moment at this station, and this moment is not relieved by gimballing.

The gravity moment in the MOD-0 is about 45,000 lb-ft peak, so its contribution to Figure 3.5 is roughly 2(45,000)(.5)(10)/62.5/5000 = 1.44 v p to p. Removing this 1P component in the flap moment would produce a 2.36 v pp value with no gimbal, and about 1.56 value with the gimbal. The loads reduction when viewed in this manner approaches 34 percent, which is closer than the theoretically maximum benefit of 50 percent, discussed in Reference 4.

Figures 3.7 and 3.8 show, as expected, that gimballing does not reduce the edgewise bending moments, although the dynamic characteristics of these moments are changed by the presence of teetering.

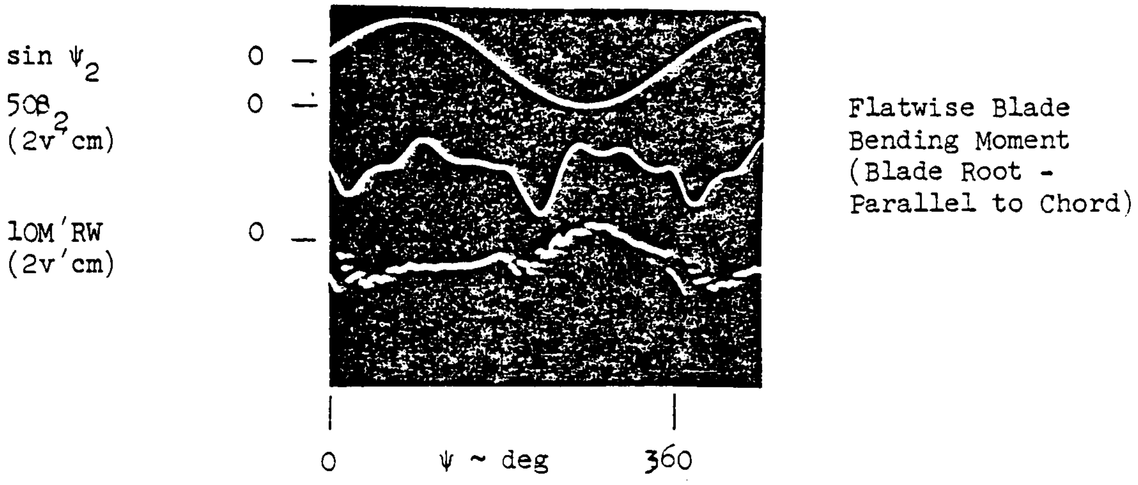


Figure 3.5 - Flatwise Moment - Nominal Conditions - With Gimbal

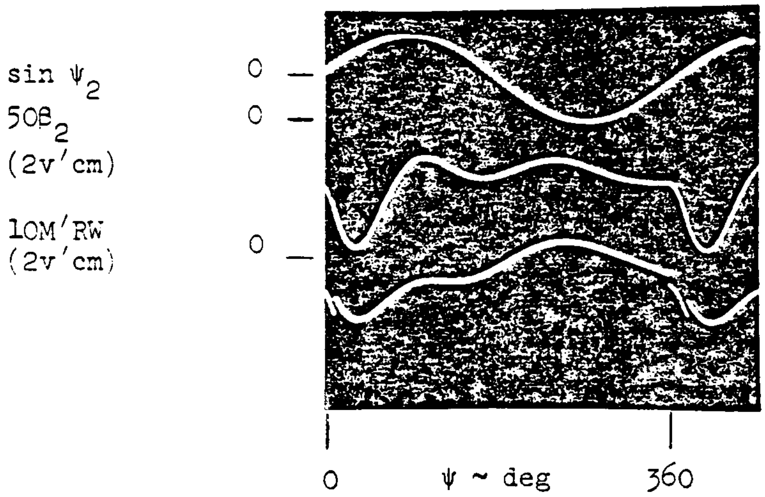


Figure 3.6 - Flatwise Moment - No Gimbal

Figures 3.5 and 3.6	REP. NO.
EFFECT OF GIMBAL SUBSYSTEM ON BLADE FLATWISE LOADS	

ORIGINAL PAGE IS
OF POOR QUALITY

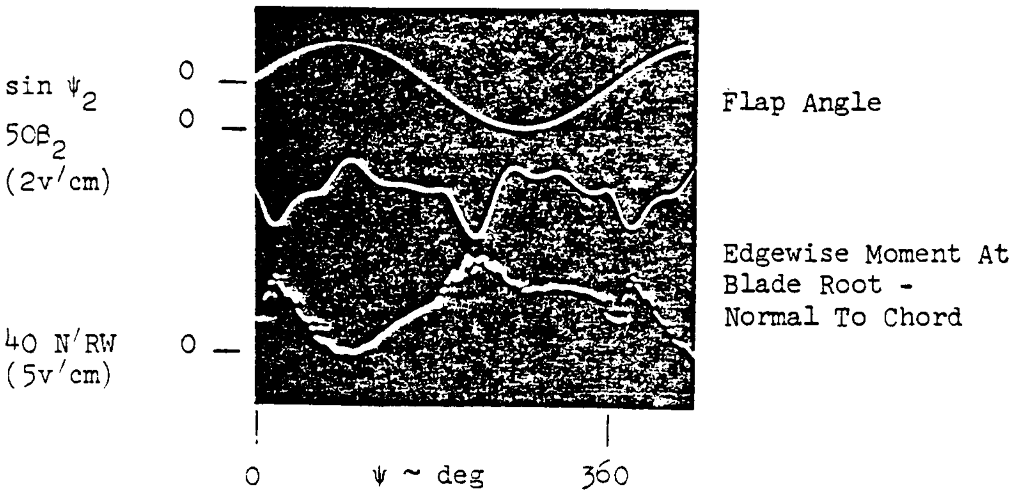


Figure 3.7 - Edgewise Moment - Nominal Conditions - Gimbal On

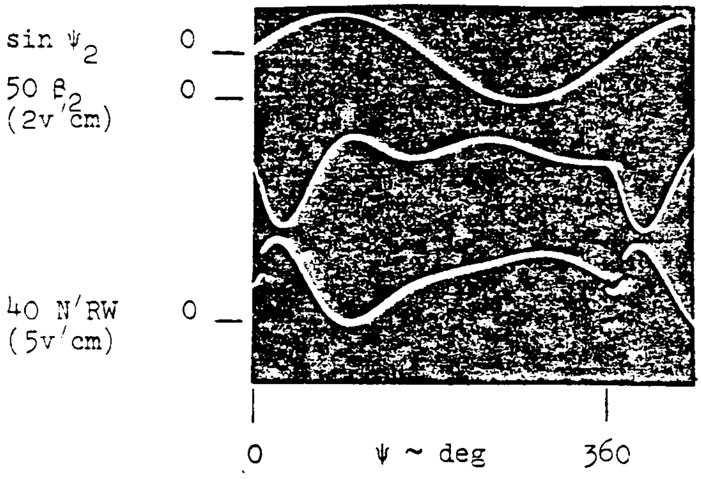


Figure 3.8 - Same As Figure 3.7 Except Gimbal Off

Figures 3.7 and 3.8	REP. NO.
EFFECT OF GIMBAL SUBSYSTEM ON BLADE EDGEWISE LOADS	

4.0 TECHNICAL PROBLEMS ENCOUNTERED IN IMPLEMENTING THE GIMBAL MODEL

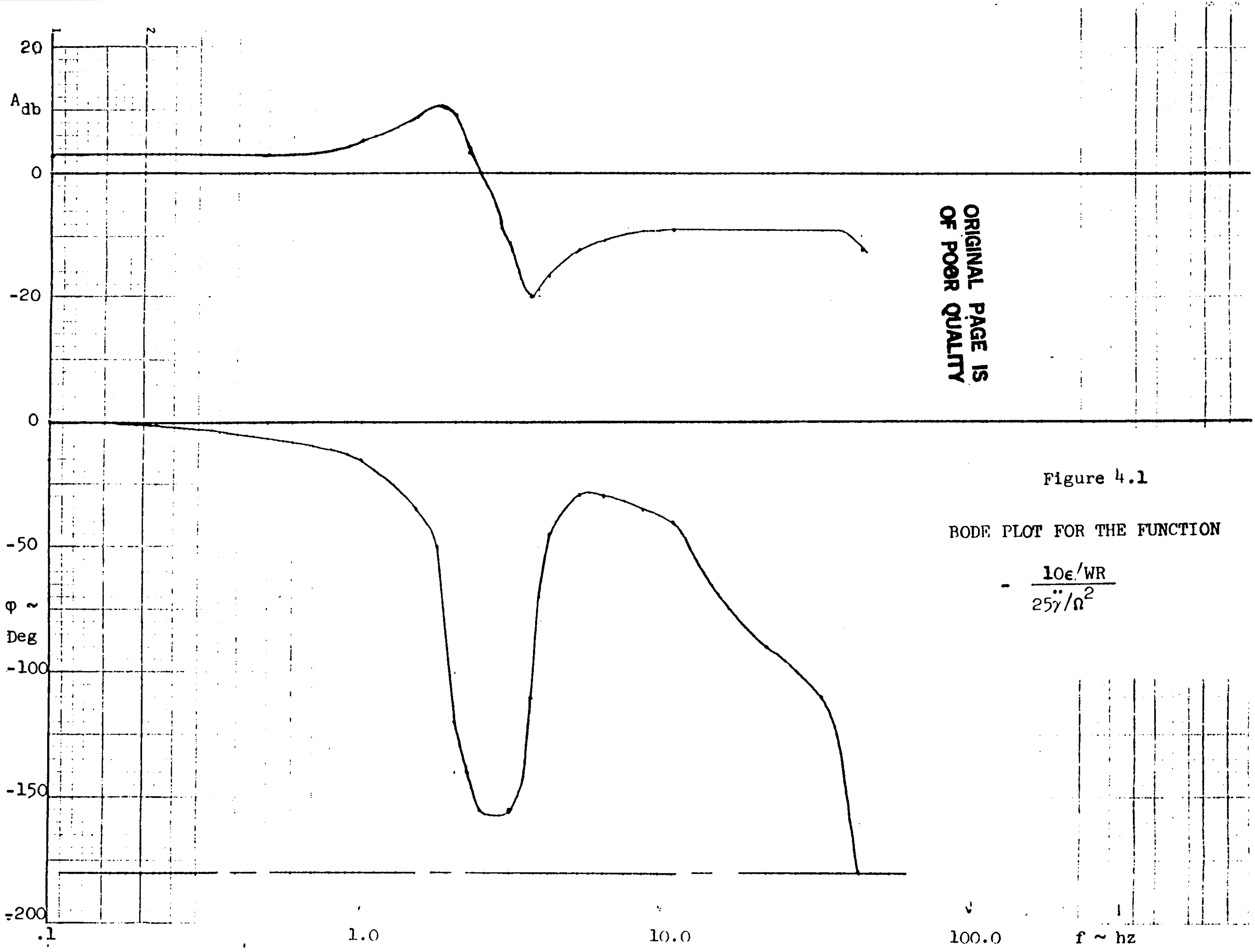
When the original system presented in Reference 2 was turned on, it burst into full oscillation. Suspecting the high-gain loop closed around $\ddot{\gamma}$ and ϵ, a Bode plot of the actual transfer function $-\epsilon/\ddot{\gamma}$, was prepared and included herein as Figure 4.1. The extreme phase shifts in the vicinity of 2hz, and at frequencies above 30 hz, reveal the sources of instability.

Neglecting the phenomenon around 2hz for a moment, one sees that a pure integrator feedback loop around the gimbal error function would be stable, if the loop gain were 10 db at 35 hz, which translates to zero db at 110.6 hz. The loop gain would then be about 44 db at .67 hz, the 1P frequency. One could expect a reduction in the 1P gimbal error of about 1/165, or down to an error of about 1/2 percent. This might be considered a maximum gimbal error reduction.

The phase lag and associated amplitude roll-off around 35 hz is because of the sample data process - the blade sweep calculations in the WEST rotor simulation. This roll-off point could be moved to a higher frequency by decreasing the sweep period from 10 ms to 1 ms in the WEST rotor. This would enable a potential gain increase of 20 db in the gimbal error loop, reducing the error to .05 percent, if this additional reduction is deemed necessary at any future time.

Notwithstanding the loop gain limit because of rotor sweep timing, the integrator feedback produces a 90 degree phase shift at all frequencies, driving the mode around 2 hz unstable at all gains above 0 db. With this phenomenon, the entire gimbal concept, as originally conceived, is obviously invalid.

To understand the mechanism associated with this phase dip, consider the simplified model of Figure 4.2. Depicted is one blade deflected



ORIGINAL PAGE IS OF POOR QUALITY

Figure 4.1

BODE PLOT FOR THE FUNCTION

$$-\frac{10\epsilon/WR}{25\ddot{\gamma}/\Omega^2}$$

in its first flap mode, with blade element mass dm, and associated differential forces.

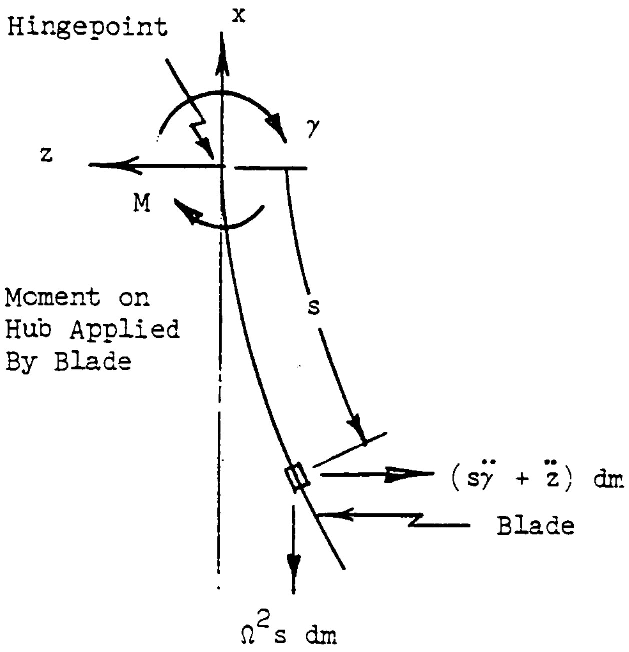

Hingepoint
x
γ
z
M
Moment on Hub Applied By Blade
s
$(s\ddot{\gamma} + \ddot{z})$ dm
Blade
$\Omega^2 s$ dm

Figure 4.2 - Simple Blade Model

From inspection of Figure 4.2, the moment applied to the hub by the blade (blade number 1) is

$$M_1 = -I\ddot{\gamma} - J(\ddot{\beta}_1 + \Omega^2\beta_1) \tag{1}$$

and, analogously, by blade number 2

$$M_2 = I\ddot{\gamma} - J(\ddot{\beta}_2 + \Omega^2\beta_2) \tag{2}$$

where the constant integrals are given by

$$I \triangleq \int m\ s^2\ ds \tag{3}$$

$$J \triangleq \int m\ s\ z_1\ ds \tag{4}$$

and the modal transformation is defined

$$z \triangleq z_1 \beta . \tag{5}$$

The blade modal equations are

$$\ddot{\beta}_1 + \omega^2 \beta_1 = -J\ddot{\gamma}/M_g \tag{6}$$

$$\ddot{\beta}_2 + \omega^2 \beta_2 = +J\ddot{\gamma}/M_g \tag{7}$$

where the generalized mass is defined

$$M_g \triangleq \int m \, z_1^2 \, ds \tag{8}$$

The gimbal error is given by the expression

$$\epsilon = M_1 - M_2 \tag{9}$$

The equations can be combined to yield the following transfer function

$$\frac{\epsilon}{\ddot{\gamma}} = \frac{k\left[\left(\frac{s}{\omega_n}\right)^2 + 1\right]}{\left(\frac{s}{\omega}\right)^2 + 1} \tag{10}$$

where

$$k \triangleq - 2(I - J^2/F^2/M_g) \tag{11}$$

$$\omega_n \triangleq \omega \sqrt{-k/2/(I-J^2/M_g)} \tag{12}$$

The Mod-0 has the following parametric values

$$M_g = 2648$$

$$\omega = 11.43 \rightarrow 1.819 \text{ hz}$$

$$I = 44{,}000$$

$$J = -10{,}094$$

$$P \triangleq \omega/\Omega = 2.728$$

$$\Omega = 4.19$$

So

$$\omega_n = 4.82 \text{ hz}$$

Function 10 has the characteristic depicted by Figure 4.1. If aerodynamics had been included in the simplified model, then damping terms would be included in the transfer function. This more general function, with the exact characteristics of Figure 4.1, has the form:

$$\frac{\epsilon}{\ddot{\gamma}} = \frac{k\left[\left(\frac{s}{\omega_n}\right)^2 + 2\xi_n\left(\frac{s}{\omega_n}\right) + 1\right]}{\left(\frac{s}{\omega}\right)^2 + 2\xi\left(\frac{s}{\omega}\right) + 1} \tag{13}$$

Figure 4.3 is a block diagram of the system that produces the characteristic of Figure 4.1. Note that the presence of $\Delta\beta = \beta_1 - \beta_2$ in the system is responsible for the dynamics. If $\Delta\beta = 0$, the transfer function would be a constant and easily controlled with pure integrator feedback.

Figure 4.3 suggests that the transfer function can be modified by inserting $\Delta\dot{\beta}$ and/or $\Delta\beta$ energy into the calculation of ϵ, phased to cancel the loops shown in Figure 4.3. Figure 4.4 shows the qualitative effects of doing this on the BODE plots. Note that both effects reduce the phase dip, improving the system phase margin.

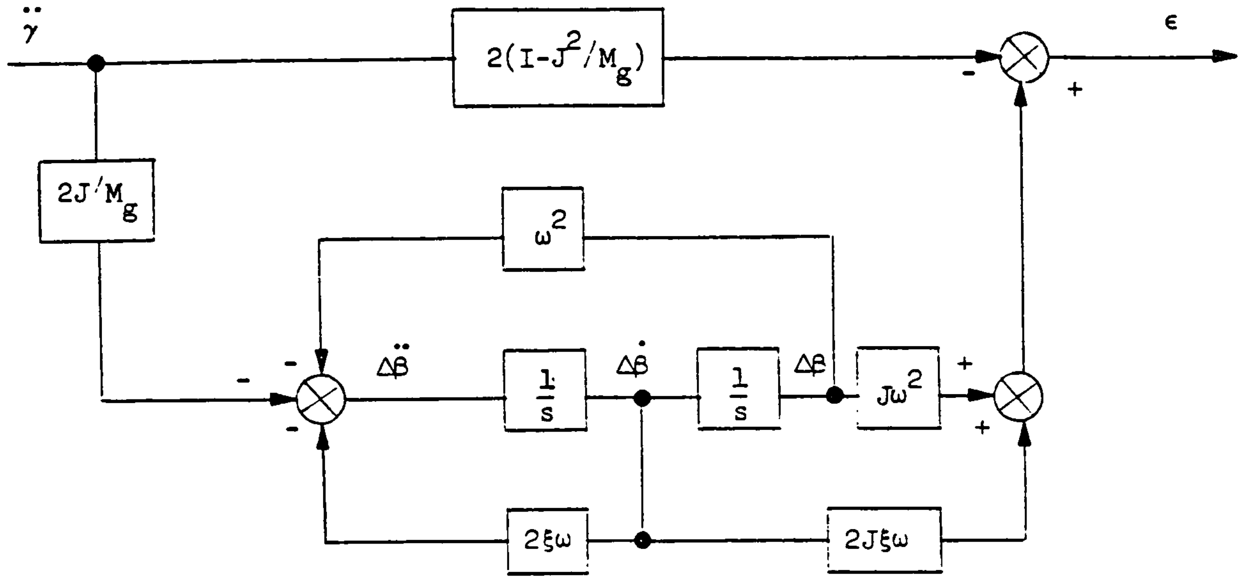


Figure 4.3 - Basic Gimbal Error Transfer Function Showing Influence of Blade Flapping Mode

The WEST rotor model has both $\Delta\beta$ and $\Delta\dot{\beta}$ signals available, using notation β_{xr} and $\dot{\beta}_{xr}$ in the program. These signals were routed to the GPURS gimbal implementation and inserted into the ϵ calculation.

Both feedbacks enhanced the stability, but the $\Delta\dot{\beta}$ was found to be superior. It did not reduce the gimbal error suppression, but did stabilize the system.

As mentioned previously, a 1P gimbal error suppression of over 1000 (over 60 db) was accomplished in the hardware, after the $\Delta\dot{\beta}$ stabilization loop was closed - this performance exceeds the 40 db maximum reduction predicted from analysis of Figure 4.1.

The final gimbal implementation diagrams are presented in Appendix A.

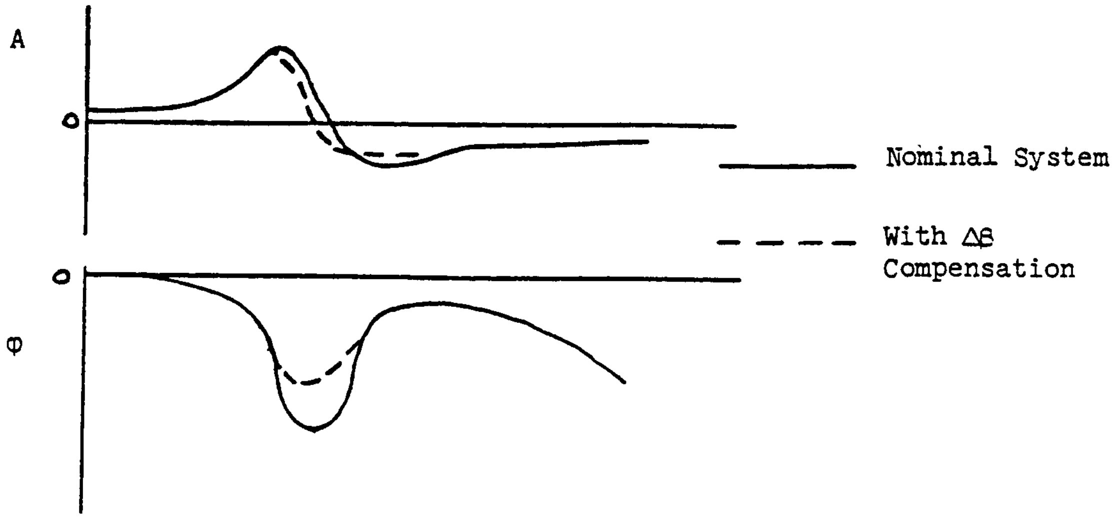


Figure 4.4a) - $\Delta\beta$ Reduces Numerator Resonance Frequency, Improving Phase Margin

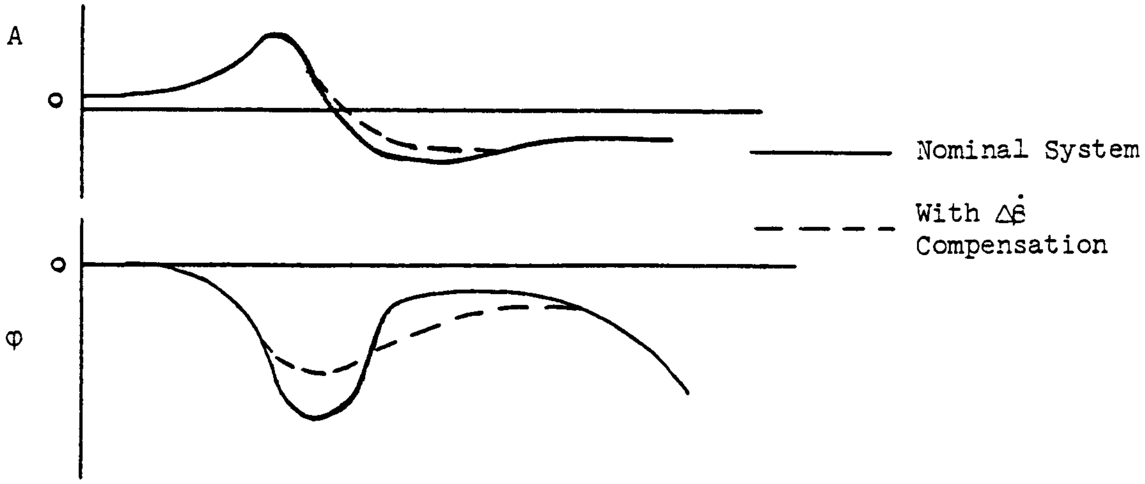


Figure 4.4b) - $\Delta\dot{\beta}$ Increases Numerator Damping Coefficient, Flattening Second Part of Dipole and Improving Phase Margin

5.0 CONCLUSIONS AND RECOMMENDATIONS

The abbreviated gimbal subsystem of Reference 1 was implemented using GPURS interfaced with WEST 2 and tested. After correcting the closed-loop stability problems that emerged, the system was tested and verified as performing well.

The next step would be to hardwire and test the full gimbal subsystem model into the WEST 2 NRS. The system should have an activate/ inactivate control so that WEST 2 will then possess the ability to analyze either teetering or hingeless wind turbine systems.

*The full system allows for shaft motion.

REFERENCES

1. Hoffman, J. A. and Thoren, R. J., "Mathematical Models and Hybrid Program for the Wind Energy System Time-Domain (WEST) Simulator - Baseline WEST Unit, PPI-1030-1, Paragon Pacific, Inc., January, 1978/revised February 20, 1979 (Rev. 6).

2. Hoffman, J. A., "Math Modelling and System Design Report - Conversion of the Wind Energy System Time-Domain Simulator, Unit 1 (WEST 1) System Design to the WEST 2 Design," PPI-1030-4, Paragon Pacific, Inc., July, 1979/Rev. 1 November 1979.

3. Hoffman, J. A., "Development Report: Wind Energy System Time-Domain (WEST) Analyzers Using Hybrid Simulation Techniques," PPI-1030-6, Paragon Pacific, Inc. for NASA Lewis Research Center, Cleveland, Ohio, NASA CR # 159737, October, 1979.

4. Hoffman, J. A., "Coupled Dynamics Analysis of Wind Energy Systems," PPI-1014-11, Paragon Pacific, Inc. for NASA Lewis Research Center, Cleveland, Ohio, NASA CR # 135152, February, 1977.

APPENDIX E

ALGEBRAIC LOOP PROBLEM IN THE POWER TRAIN MODEL

This Appendix presents an example derivation of a single degree-of-freedom power train mathematical model. A significant origin of numerical instability is identified herein, and the reformulation methods needed to correct the problem are discussed. These same reformulation procedures are used elsewhere in this report in derivations of more general power train mathematical models.

2.3 Power Train/Power Sink Model

Figure 2-3 shows the single dof power train model, in conjunction with an effective rotor inertia J_R. N_H is the rotor torque produced by the WEST rotor model. Ω and $\dot{\Omega}$ are rotor speed and acceleration variables, which must be produced by the power train model and applied to the rotor model as inputs. γ_P is the torque applied to the power train by friction and the power-absorbing elements (i.e., the alternator) of the wind energy system. Note that the power train subsystem has the simple equation

$$N_H = -J_P\dot{\Omega} - \gamma_P \tag{2-6}$$

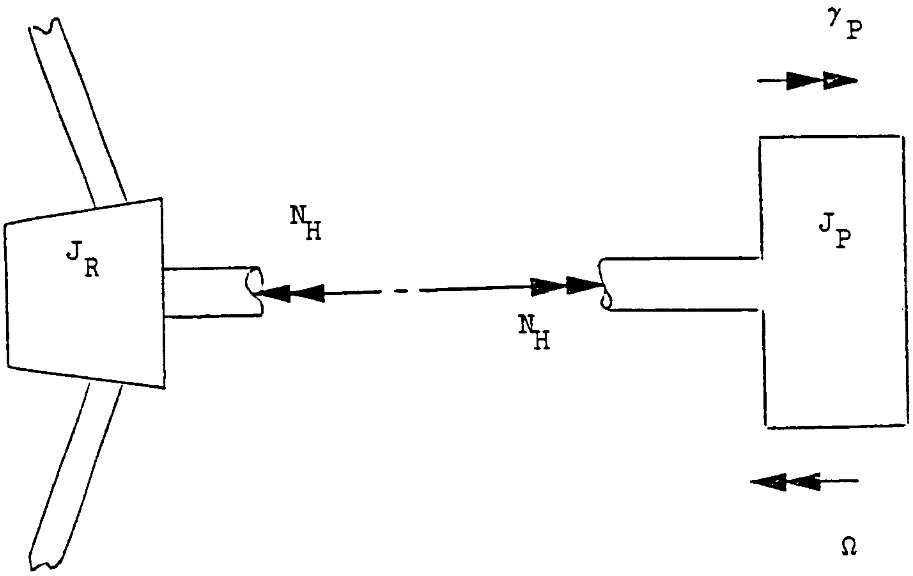


Figure 2-3 - Power Train Single DOF Model

Figure 2-4 is a block diagram of the rotor/power train system, showing the required Ω and $\dot{\Omega}$ feedbacks.

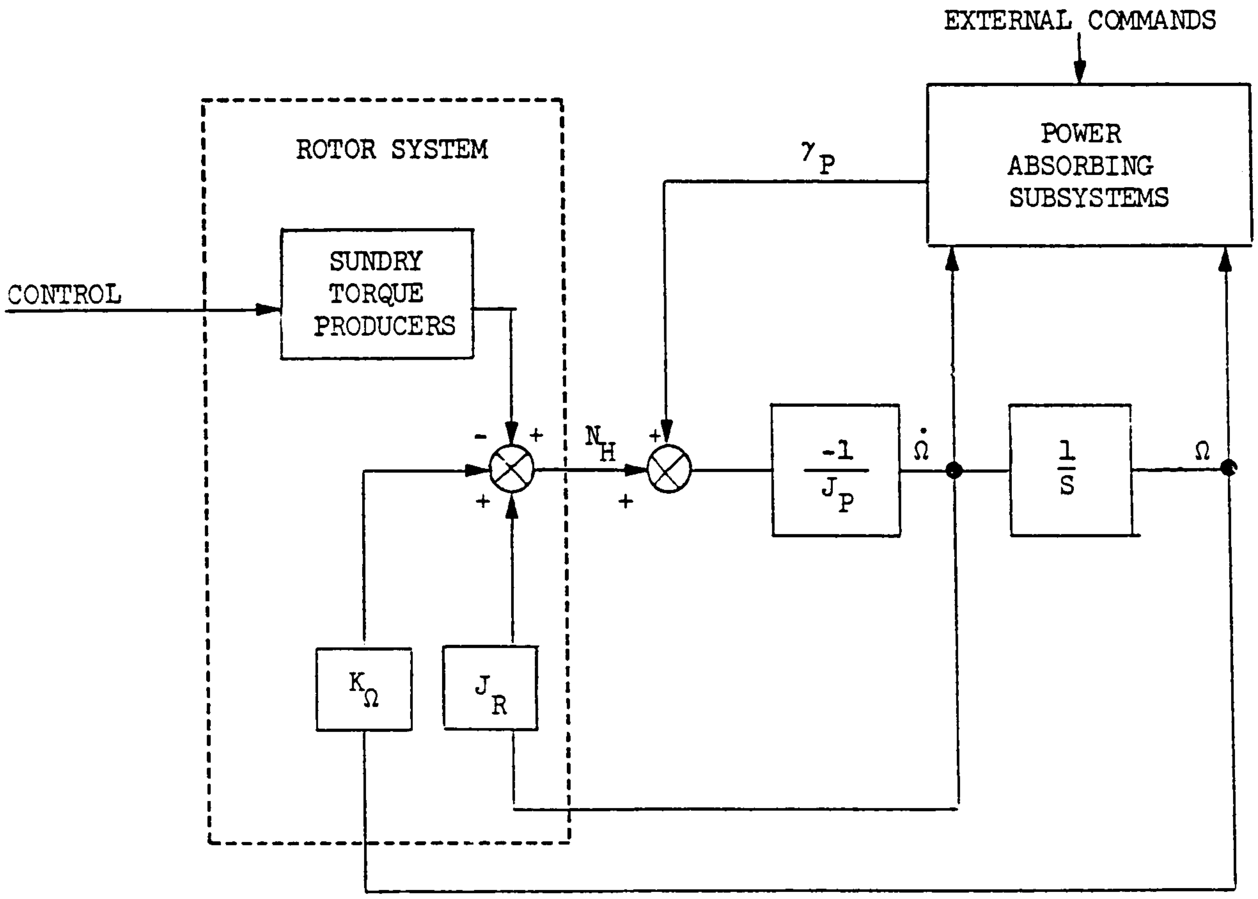


Figure 2-4 - Generic Power Train Model
(1 DOF) With Rotor Couplings

On inspection of Figure 2-4, one notes the "algebraic" loop from $\dot{\Omega}$ through J_R, N_H, $1/J_P$ and back to $\dot{\Omega}$ again. Such loops can produce high frequency instabilities in multiplexed systems such as the WEST rotor model; rearranging the configuration into a form depicted by Figure 2-5 usually eliminates the problem, however, by effectively reducing the loop gain from $1/J_P$ to $1/(J_P + J_R)$. A high frequency low-pass filter with time constant τ has also been added, to remove high frequency energy from the loop and further enhance stability. τ is chosen to be small enough (e.g., 2-10 ms), so that the filter's presence does not aberrate the accuracy of the calculation in the frequency band of interest (usually below 10 Hz).

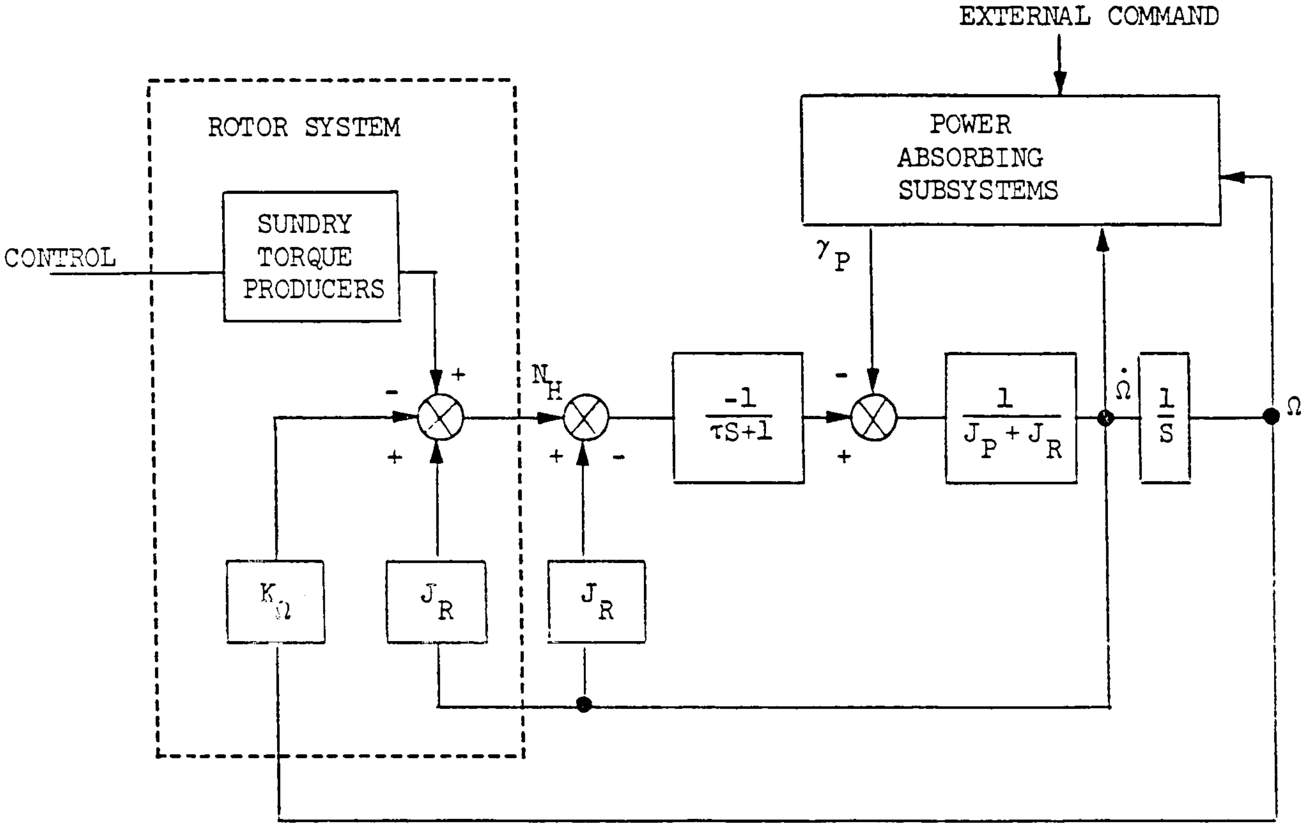


Figure 2-5 - Alternative Power Train Implementation

The system of Figure 2-5 is shown in its detailed implementation form in Section 4. The power absorbing equation has been implemented in the form

$$\gamma_P = k_d\Omega + k_P \int(\Omega - \Omega_n)\,dt + \gamma_{Po} \qquad (2\text{-}7)$$

The term with the k_d factor is a damping term, which can also be used to simulate alternators whose torques are related to speed. k_P is a "phase" gain, which simulates alternators that produce torque in proportion to a phase difference between the angle $\int\Omega dt$ and some network phase $\int\Omega_n dt$. γ_{Po}, of course, is a steady bias torque associated with some nominal power setting

APPENDIX F

VARIABLES AND CONSTANTS: DATA BASE REPRESENTATION

Presented in this Appendix are the definitions, including normalization and scale factors, of all the variables and constants used in the wind turbine simulation model.

```
C
C  VARIABLES APPEARING IN THE SYSTEM EQUATIONS
C
C  JUNE 1985
C
C  1. THE SIMULATION MODEL IS SET UP FOR THE FOLLOWING MAXIMUMS:
C
C        BLADES .................................(NB)   2
C        RADIAL STATIONS PER BLADE ..............(NR)  13
C        CONTROL SYSTEM STATE VECTOR ELEMENTS ....(NC)   3
C        ELASTIC MODES PER BLADE .................(NM)   3
C        GIMBAL DEGREES OF FREEDOM ...............(NG)   1
C        POWER TRAIN DEGREES OF FREEDOM ..........(NP)   2
C
C  2. THE FOLLOWING CONSTANTS ARE USED FOR NORMALIZATION AND SCALING:
C
C        C34    BLADE CHORD AT 3/4 RADIUS (FEET)
C        FR     REFERENCE FORCE (LBS)
C        MR     REFERENCE MOMENT = R*FR (FT-LBS)
C        NB     NUMBER OF BLADES
C        OR     REFERENCE ROTATIONAL VELOCITY = ZO
C        R      ROTOR RADIUS (FEET)
C        RAR    REFERENCE ROTATIONAL ACCELERATION = ZO**2
C        VR     REFERENCE TRANSLATIONAL VELOCITY = R*ZO
C        XAR    REFERENCE TRANSLATIONAL ACCELERATION = R*ZO**2
C        ZO     NOMINAL ROTOR SPIN RATE (RAD/SEC)
C
C
C
C  ROTOR MODEL : 1.1 THROUGH 1.17
C
     BOO(1)+2 : 1.,5.,5. : 1/R+2
     QUIESCENT BLADE SHAPE FUNCTION S:B,B:S,R:S

     B(1)+2 : 1.,5.,5. : 1./R+2
     POSITION OF BAX WRT SAX. S:B,B:S,R:S.

     BD(1)+2 : 5.,2.5,2.5 : 1./VR+2
     VELOCITY OF BAX WRT SAX. S:B,B:S,R:S

     BDD(1)+2 : 10.+2 : 1./XAR+2
     ACCELERATION OF BAX WRT SAX. S:B,B:S,R:S

     BF(1)+2 : 5.+2 : 1./R+2
     TRANSLATIONAL PERTURBATION OF BRL DUE TO FEATHERING T:XP,S:B,B:S,R:S.

     BP(1)+2 : 10.,5.,5. : 1./R+2
     PERTURBATION OF BRL FROM QUIESCENT POSITION, BP=B-BOO.

     CAB : 1. : 1.
     COSINE OF BLADE SECTION ANGLE OF ATTACK (WRT CHORDLINE)

     CB : .5 : 1./C34
     LOCAL BLADE CHORD,FT.
```

CFA2 : .5 : 1.
BLADE SECTION DRAG COEFFICIENT.(USED IN AERO PACKAGE)

CFA3 : .5 : 1.
BLADE SECTION LIFT COEFFICIENT.(USED IN AERO PACKAGE)

CFS : 1. : 1.
COS(FS); FS = BLADE FEATHERING ANGLE.

CMA : 10. : 1.
BLADE SECTION MOMENT COEFFICIENT.(USED IN AERO PACKAGE)

CY : 1. : 1.
COS(Y); Y = BLADE # 1 AZIMUTHAL ANGLE.

CYBI : 1. : 1.
COS(YBI); YBI = ANGLE BETWEEN BLADE # 1 AND BLADE # I.

CYBION : 1. : 1.
CYBI/NB

CYI : 1. : 1.
COS(Y+YBI); (Y+YBI) = BLADE # I AZIMUTHAL ANGLE.

DA(1)1+2,DA(1)2+2,DA(1)3+2 : 1.+8 : 1.+8
ROTATIONAL MODESHAPE MATRIX - REPEATS FOR EVERY RADIAL POSITION, J.

DB(1)1+2,DB(1)2+2,DB(1)3+2 : 5.,2.5+1,5.,2.5+1,5.,2.5+1 : 1./R+8
TRANSLATIONAL MODESHAPE MATRIX - REPEATS FOR EVERY RADIAL POSITION, J.

DRAG : .25 : NB*R/FR
AERODYNAMIC DRAG MAGNITUDE.

DS : 10. : 1./R
BLADE SEGMENT LENGTHS USED IN RADIAL SUMMATIONS

EA(1)+2 : 2.+2 : 1.+2
ROTATIONAL BRL MOVEMENT DUE TO BLADE MODES - EULER ANGLES

EB(1)+2 : 10.,5.,5. : 1./R+2
TRANSLATIONAL BRL MOVEMENT DUE TO BLADE MODES.

EDA(1)+2 : 1.+2 : 1./OR+2
TIME DERIVATIVE OF EA

EDB(1)+2 : 5.,2.5,2.5 : 1./VR+2
TIME DERIVATIVE OF EB.

EDDA(1)+2 : .2+2 : 1./RAR+2
TIME DERIVATIVE OF EDA

EDDB(1)+2 : 1.,.5,.5 : 1./XAR+2
DOUBLE TIME DERIVATIVE OF EB.

EG1 : .25 : 1./MR

GIMBAL ERROR FUNCTION

F : .5 : 1.
BLADE FEATHERING CONTROL ANGLE, RAD.

FAO(1)+2 : .5,.25,.1 : NB*R/FR+2
QUIESCENT DISTRIBUTED AERO LOADING FORCE, LB/FT. R:S.

FABC(2)+1 : .5,.2 : NB*R/FR+1
AERO DISTRIBUTED FORCE COLUMN, R:BAX.

FBLD(1)+2 : 1.,1.,.1 : NB/FR+2
BLADE # 1 ROOT FORCES, RESOLVED IN SHANK AXES

FG(1)+2 : 1.,1.,.1 : 1./FR+2
FORCES APPLIED BY ALL BLADES TO GIMBAL RING. R:G.

FGR(1)+2 : 1.,1.,.1 : 1./FR+2
FORCES SUMMED FOR ALL BLADES, RESOLVED TO GAX.

FH(1)+2 : 1.,1.,.2 : 1./FR+2
FORCES APPLIED TO SUPPORTS BY ROTOR, RESOLVED TO HAX.

FMA(1)+2 : .5,.25,.1 : NB*R/FR+2
MODIFIED DISTRIBUTED AERO FORCE COLUMN

FMB(1)+2 : .5,.25,.1 : NB*R/FR+2
FMA+FMI.

FMI(1)+2 : .5,.25,.1 : NB*R/FR+2
MODIFIED DISTRIBUTED INERTIAL FORCE COLUMN

FP(1)1+2,FP(1)2+2,FP(1)3+2 : .2+2,.4+2,1.,1.,.5 : 1.+8
SHANK FORCE INTEGRAND PREMULTIPLIER TENSOR

FRI(1)+2 : 1.,1.,.1 : NB/FR+2
BLADE ROOT FORCES APPLIED TO GIMBAL RING FOR BLADE NO. I. R:S.

FS : .5 : 1.
F-FO (FEATHERING ANGLE)

FSO1,FSO2,FSO3 : .1,1.,.1 : 1./FR+2
BLADE ROOT FORCES IN SHANK FORCE FORMULA (QUIESCENT STATE).

FSQ1(1)+2,FSQ2(1)+2,FSQ3(1)+2 : .5+2,.5,.125+1,.05+2 : 1./FR+8
OPERATOR IN SHANK FORCE FORMULA.

FSQD1(1)+2,FSQD2(1)+2,FSQD3(1)+2 : 1.+2,1.+2,.1+2 : OR/FR+8
OPERATOR IN SHANK FORCE FORMULA.

GI(1)+2 : .25+2 : 1./RAR+2
GENERALIZED FORCING FUNCTION COLUMN FOR Ith BLADE.

GP(1)1+2,GP(1)2+2,GP(1)3+2 : .05+2,.1+2,.25+2 : FR/NB*RAR+8
GENERALIZED FORCING FUNCTION INTEGRAND PREMULTIPLIER TENSOR.

GP(1)4+2 : 1.,1.,.8 : FR/NB*RAR+1,FR*C34/NB*R*RAR
GENERALIZED FORCING FUNCTION INTEGRAND PREMULTIPLIER TENSOR.

GRI(1)+2 : .25+2 : 1./RAR+2
GENERALIZED FORCING FUNCTION OVER GENERALIZED MASS-ROTOR BLADE MODES.

IFR(1)+2 : .1,.1,.1 : NB*R/FR+2
INTEGRAND FUNCTIONS-BLADE ROOT(GIMBAL RING) LOADS. R : R.

IGR(1)+2 : .025+2 : R/RAR+2
INTEGRANDS - BLADE GENERALIZED FORCING FUNCTION RADIAL INTEGRATION.

IMR(1)+2 : .1,.25,.25 : NB/FR+2
INTEGRAND FUNCTIONS-BLADE ROOT(GIMBAL RING) LOADS. R:R.

IX(1)+2 : 4.,8.+1 : 1.+2
INTEGRAND FOR X FUNCTION.

IXD(1)+2 : 2.,1.,1. : 1./ZO+2
INTEGRAND FOR XD FUNCTION

IXDD(1)+2 : .5,1.,1. : 1./RAR+2
INTEGRAND FOR XDD FUNCTION.

LIFT : .1 : NB*R/FR
AERODYNAMIC LIFT MAGNITUDE.

M : 2. : 1.
BLADE SECTION MACH NUMBER (USED IN AERO PACKAGE).

M(1)+2 : 5.,2.,5. : 1.+2
MACH NUMBER COMPONENTS RESOLVED TO BAX.

MB : .025 : NB*VR*VR/FR
DISTRIBUTED BLADE MASS, SLUGS/FT.

MBLD(1)+2 : .25,.25,1. : NB/MR+2
BLADE # 1 ROOT MOMENTS, RESOLVED IN SHANK AXES

MG(1)+2 : .25,.25,1. : 1./MR+2
MOMENTS APPLIED BY ALL BLADES TO GIMBAL RING. R:G.

MGR(1)+2 : .25,.25,1. : 1./MR+2
MOMENTS SUMMED FOR ALL BLADES, RESOLVED TO GAX.

MH(1)+2 : .25,.25,1. : 1./MR+2
MOMENTS APPLIED TO SUPPORTS BY ROTOR, RESOLVED TO HAX.

MP(1)1+2,MP(1)2+2,MP(1)3+2 : .2,.5,.5,.4,1.+1,1.,1.25,2.5 : 1./R+8
SHANK MOMENT INTEGRAND PREMULTIPLIER TENSOR.

MRI(1)+2 : .25,.25,1. : NB/MR+2
BLADE ROOT MOMENTS APPLIED TO GIMBAL RING FOR BLADE NO. I. R:S.

MS01,MS02,MS03 : .25,.25,1. : 1./MR+2
BLADE ROOT MOMENTS IN SHANK MOMENT FORMULA (QUIESCENT STATE).

MSQ : 1. : 1.
MACH NUMBER, M, SQUARED (USED IN AERO PACKAGE).

MSQ1(1)+2,MSQ2(1)+2,MSQ3(1)+2 : .125+2,.125+2,.5,.25+1 : 1./MR+8
OPERATOR IN SHANK FORCE FORMULA

MSQD1(1)+2,MSQD2(1)+2,MSQD3(1)+2 : .25+2,.25+2,1.+2 : OR/MR+8
OPERATOR IN SHANK FORCE FORMULA

QO(1)+2 : 2.+2 : 1.+2
STEADY COMPONENT OF MULTIBLADE MODAL COORDINATES

QAC : .0625 : NB*R/FR
AERODYNAMIC PRESSURE TIMES LOCAL BLADE CHORD.

QC(1)+2 : 2.+2 : 1.+2
COSINE COMPONENT OF MULTIBLADE MODAL COORDINATES

QDO(1)+2 : 1.+2 : 1./OR+2
STEADY COMPONENT OF MULTIBLADE MODAL VELOCITY

QDC(1)+2 : 1.+2 : 1./OR+2
COSINE COMPONENT OF MULTIBLADE MODAL VELOCITY

QDDG1 : 5. : 1./RAR
DOUBLE TIME DERIVATIVE OF GIMBAL DEFLECTION ANGLE.

QDDI(1)+2 : .2+2 : 1./RAR+2
DOUBLE TIME DERIVATIVE OF BLADE I MODAL COORDINATES.

QDG1 : 5. : 1./OR
TIME DERIVATIVE OF GIMBAL DEFLECTION ANGLE.

QDI(1)+2 : 1.+2 : 1./OR+2
TIME DERIVATIVES OF BLADE I MODAL COORDINATES.

QDS(1)+2 : 1.+2 : 1./OR+2
SINE COMPONENT OF MULTIBLADE MODAL VELOCITY

QG1 : 5. : 1.
GIMBAL DEFLECTION ANGLE

QI(1)+2 : 2.+2 : 1.+2
BLADE MODAL COORDINATES FOR BLADE NO. I.

QS(1)+2 : 2.+2 : 1./NB+2
SINE COMPONENT OF MULTIBLADE MODAL COORDINATES

QSG1 : 5. : 1.
SAVE GIMBAL ANGLE

RO(1)1+2,RO(1)2+2,RO(1)3+2 : 1.,2.+1,2.,.5,1., 2.,1.,.5 : 1.+8

ROTATION MATRIX - SAX TO BAX CONTROL POSITION.

ROO(1)1+2,ROO(1)2+2,ROO(1)3+2 : 1.,2.+2,1.+1,2.,1.+1 : 1.+8
ROTATION MATRIX, SAX TO BAX - BLADE IN QUIESCENT POSITION.

R(1)1+2,R(1)2+2,R(1)3+2 : 1.,2.+1,2.,.5,1., 2.,1.,.5 : 1.+8
SHANK TO BLADE AXES EULERIAN X - FORMATION MATRIX, PACKED BY COLUMN.

RFS : 1. : 1.
OPERATOR ON F-FO TO PRODUCE NO. 1 SAX TO BAX ROTATION, CONTROL POSITION

RR1(2)+1 : 2.+1 : 1.+1
ROTATION MATRIX ELEMENTS - RELATIVE WIND TO BAX

RR2(2)+1 : 1.,2.5 : 1.+1
ROTATION MATRIX ELEMENTS - RELATIVE WIND TO BAX

RR3(2)+1 : 1.+1 : 1.+1
ROTATION MATRIX ELEMENTS - RELATIVE WIND TO BAX

S : 1. : 1./R
LINE COORDINATE OF BLADE RADIAL STATION ALONG BRL

SAB : 1. : 1.
SINE OF BLADE SECTION ANGLE OF ATTACK (WRT CHORDLINE)

SFS : 1. : 1.
SIN(FS); FS = BLADE FEATHERING ANGLE.

SY : 1. : 1.
SIN(Y); Y = BLADE # 1 AZIMUTHAL ANGLE.

SYBI : 1. : 1.
SIN(YBI); YBI = ANGLE BETWEEN BLADE # 1 AND BLADE # I.

SYBION : 1. : 1.
SYBI/NB.

SYI : 1. : 1.
SIN(Y+YBI); (Y+YBI) = BLADE # I AZIMUTHAL ANGLE.

UMBR(1)+2 : 10.,10.,2.5 : 1./XAR+2
BLADE ELEMENT MODIFIED ACCELERATION, R : SAX.

US(1)+2 : 10.+2 : 1./XAR+2
ACCELERATION OF SHANK AXES. T:XA,S:S,B:I,R:S

VA(1)+2 : 2.5,.5,2.5 : 1./VR+2
AERO VELOCITY OF BLADE AXES. S:B,B:A,R:B

VAA(1)+2 : 2.5,.5,2.5 : 1./VR+2
AUGMENTIVE AERO VELOCITY DUE TO INTERFERENCE EFFECTS

VAC(1)+2 : 2.5+2 : 1./VR+2
INTERMEDIATE VARIABLE

VACT(1)+2 : 2.5,2.5,2.5 : 1./VR+2
AERO VELOCITY SEEN BY THE BLADE (INCLUDING INTERFERENCE EFFECTS)

VAH(1)+2 : 2.5+2 : 1./VR+2
AIRSPEED OF HAX, S:H,B:A,R:H

VAS(1)+2 : 2.5+2 : 1./VR+2
SHANK AERODYNAMIC VELOCITY OF BLADE I. S:S,B:A,R:S.

VATU(1)+2 : 2.5+2 : 1./VR+2
INTERMEDIATE VARIABLE

VC(1)+2 : 20.+2 : 1./VR+2
INTERMEDIATE VARIABLE

VDC(1)+2 : 10.+2 : 1./XAR+2
INTERMEDIATE VARIABLE

VDH(1)+2 : 10.+2, : 1./XAR+2
TIME DERIVATIVE OF VH TAKEN WRT HAX, S:H,B:I,R:H.

VDS(1)+2 : 10.+1,20. : 1./XAR+2
TIME DERIVATIVE OF SHANK INERTIAL VELOCITY WRT SAX. S:S,B:I,R:S,D:S

VDTU(1)+2 : 10.+2 : 1./XAR+2
INTERMEDIATE VARIABLE

VH(1)+2 : 20.+2 : 1./VR+2
INERTIAL VELOCITY OF HAX, S:H,B:I,R:H

VS(1)+2 : 20.+2 : 1./VR+2
INERTIAL VELOCITY OF BLADE SHANK AXES. S:S,B:I,R:S.

VTU(1)+2 : 20.+2 : 1./VR+2
INTERMEDIATE VARIABLE

VWT(1)+2 : 2.5+2 : 1./VR+2
WIND VELOCITIES INCLUDING TURBULENCE AND RETARDATION.

WAC(1)+2 : 2.5+1,.5 : 1./OR+2
INTERMEDIATE VARIABLE

WAH(1)+2 : 2.5+2 : 1./OR+2
AIRSPEED OF HAX, S:H,B:A,R:H

WAS(1)+2 : 2.5+2,.5 : 1./OR+2
SHANK AERODYNAMIC VELOCITY OF BLADE I. S:S,B:A,R:S.

WATU(1)+2 : 2.5+1,.5 : 1./OR+2
INTERMEDIATE VARIABLE

WCMZ(1)+2 : 5.,5.,10. : 1./OR+2
INTERMEDIATE VARIABLE, (WC-Z)

```
WDC(1)+2 : 2.5+1,5. : 1./RAR+2
INTERMEDIATE VARIABLE

WDH(1)+2 : 2.5,2.5,5. : 1./RAR+2
TIME DERIVATIVE OF WH TAKEN WRT HAX, S:H,B:I,R:H.

WDS(1)+2 : 2.5+1,5. : 1./RAR+2
TIME DERIVATIVE OF SHANK INERTIAL VELOCITY WRT SAX. S:S,B:I,R:S,D:S

WDSI(1)+2 : 2.5+2 : 1./RAR+2
TIME DERIVATIVE WSI TAKEN WRT SAX. S:S,B:I,R:S,D:S

WDTU(1)+2 : 2.5+1,5. : 1./RAR+2
INTERMEDIATE VARIABLE

WH(1)+2 : 5.,5.,10. : 1./OR+2
INERTIAL VELOCITY OF HAX, S:H,B:I,R:H

WS(1)+2 : 5.,5.,.5 : 1./OR+2
INERTIAL VELOCITY OF BLADE SHANK AXES. S:S,B:I,R:S.

WSMZ(1)+2 : 5.+1,10. : 1./OR+2
WS MINUS Z, WS WITH Z REMOVED TO IMPROVE SCALING

WTUMZ(1)+2 : 5.,5.,10 : 1./OR+2
INTERMEDIATE VARIABLE, ( WTU-Z )

WWT(1)+2 : 2.5+2 : 1./OR+2
WIND VELOCITIES INCLUDING TURBULENCE AND RETARDATION.

X(1)+2 : 40.+2 : 1./R+2
PERTURBATION OF BRL FROM POSITION CALCULATED USING LINEAR MODEL.

XD(1)+2 : 20.+2 : 1./VR+2
PERTURBATION OF BRL SPEED FROM THAT COMPUTED USING LINEAR MODEL.

XDD(1)+2 : 10.+2 : 1./XAR+2
PERTURBATION OF BRL ACCELERATION FROM THAT COMPUTED USING LINEAR MODEL.

Z : .5 : 1./OR
ROTOR SPIN RATE-RAD/SEC

ZD : 5. : 1./RAR
ROTOR SPIN ACCELERATION-TIME DERIVATIVE OF Z.

ZP : 1. : 1.
(Z/ZO - 1.), PERTURBATION OF Z FROM THE REFERENCE VALUE, ZO.

ZSQ : .25 : 1./RAR
Z SQUARED.

ZZ : .5 : 1.
(Z/ZO)**2 - 1 , QUADRATIC DIFFERENTIAL SPIN RATE.

C
```

```
C TOWER SHADOW AND WIND SHEAR
C
C FOR CONVENIENCE, THESE EFFECTS HAVE BEEN INCLUDED IN THE ROTOR MODEL.
C NORMALLY THEY ARE PART OF THE "AIR MASS" SUBSYSTEM.
C
      SHD : 1. : 1.
      SHADOW EFFECT-MODULATING COEFFICIENT.

      SHR : 1. : 1.
      WINDSHEAR MODULATING COEFFICIENT
```

```
C
C SUPPORT SYSTEM : 2.0
C
      GS(1)+1 : .5+2 : 1./RAR+2
      GENERALIZED FORCING FUNCTION ON SUPPORT SYSTEM MODES

      MPOD : 1. : 1./MR
      MOMENT ABOUT THE 3-AXIS TRANSMITTED FROM THE POD TO THE TOWER

      S(1)+2 : 10.,10.,20. : 1.+2
      SUPPORT SYSTEM MODAL COORDINATES

      SD(1)+2 : 1.,1.,2. : 1./OR+2
      TIME DERIVATIVE OF SUPPORT SYSTEM MODAL COORDINATES.

      SDD(1)+1 : .5,.5,.5 : 1./RAR+2
      SECOND TIME DERIVATIVE OF SUPPORT SYSTEM MODAL COORDINATES.

      VBH(1)+2 : 20.+2, : 1./VR+2
      INERTIAL VELOCITY OF TOWER BASE, S:BT,B:I,R:H.

      VDBH(1)+2 : 10.+2 : 1./XAR+2
      TIME DERIV. OF INERTIAL VELOC. TAKEN WITH RESPECT TO TOWER BASE AXES.

      WBH(1)+2 : 5.,5.,10. : 1./OR+2
      INERTIAL VELOCITY OF TOWER BASE, S:BT,B:I,R:H.

      WDBH(1)+2 : 2.5+1,5. : 1./RAR+2
      TIME DERIV. OF INERTIAL VELOC. TAKEN WITH RESPECT TO TOWER BASE AXES.

C
C CONTROL SYSTEM : 3.0
C
      A(1)+2 : 2.,.5,5. : 1.+2
      CONTROL SYSTEM STATE VECTOR.

      AD(1)+2 : 2.,.02,.5 : 1./OR+2
      TIME DERIVATIVE OF A.

      C(1)+2 : .5,5.+1 : 1.+2
      ROTOR CONTROLS, COLLECTIVE PITCH, A1 AND B1 CYCLIC PITCH(RADIANS).

      LGC : 1. : 1./MR
      ELECTRICAL GENERATOR/ALTERNATOR CONTROL.

      ZA : .5 : 1./OR
      ALTERNATOR/GENERATOR ROTOR ROTATIONAL SPEED.

      ZERR : 2. : 1./OR
      ERROR IN THE RATIOS OF ROTOR SPEED: (ZREQ/ZO - Z/ZO)
```

```
C
C POWER TRAIN : 4.0
C
      CDY : .5 : 1.
      ONE HALF OF COSINE OF ROTOR ADVANCE ANGLE, COS(DY)

      CYHAF : .5 : 1.
      ONE HALF OF COSINE OF THE ROTOR AZIMUTHAL ANGLE CY.

      CYSAV : .5 : 1.
      HOLDS A "SAVE" VERSION OF CYHAF.

      EA : .25 : 1.
      AMPLITUDE ERROR IN RATE RESOLVER - POWER TRAIN MODEL.

      P1,P2 : 1.,25. : 1.+1
      POWER TRAIN COORDINATE VECTOR

      PD1,PD2 : .5,5. : 1./OR+1
      TIME DERIVATIVE OF POWER TRAIN COORDINATES,P.

      PDD1,PDD2 : 5.,.5 : 1./RAR+1
      DOUBLE TIME DERIVATIVE OF POWER TRAIN COORDINATES,P.

      SDY : 1. : 1.
      SINE OF ROTOR ADVANCE ANGLE, SIN(DY).

      SYHAF : .5 : 1.
      ONE HALF OF SINE OF THE ROTOR AZIMUTHAL ANGLE SY.

C
C AIR MASS : 5.0
C
      EVR : .04 : 1.
      ERROR FUNCTION ON VELOCITY RECIPROCAL IN AIR MASS MODELS.

      QGU(1)+5 : 256.+5 : 1.+5
      WIND GUST FILTER COLUMN OF STATISTICAL DEGREES OF FREEDOM.

      QDGU(1)+5 : 16.+5 : 1./OR+5
      TIME DERIVATIVE OF QG (WINDGUST STATISTICS).

      QDDGU(1)+5 : 1.+5 : 1./RAR+5
      DOUBLE TIME DERIVATIVE OF QG(WINDGUST STATISTICS).

      QDGUO(1)+5 : 16.+5 : 1./OR+5
      OLD GUST VELOCITY USED IN NUMERICAL INTEGRATION.

      RANDOM : 1. : 1.
      RANDOM VARIABLE (RANDOM NUMBER) IN WINDGUST MODEL.

      RV : .04 : VR
      VELOCITY RECIPROCAL IN AIR MASS RETARDATION MODEL.

      VG(1)+2 : 2.5+2 : 1./VR+2
      WINDGUST VELOCITIES.

      VRW : 2.5 : 1./VR
      WIND RETARDATION VELOCITY-AIR MASS MODELS.

      VW(1)+2,WW(1)+2 : 2.5+5 : 1./VR+2,1./OR+2
      WIND INERTIAL VELOCITY. S:W,B:I,R:O

      WG(1)+2 : 2.5+2 : 1./OR+2
      WINDGUST VELOCITIES.

      WW(1)+2 : 2.5+2 : 1./OR+2
      WIND INERTIAL VELOCITY. S:W,B:I,R:O
C
C TOWER SHADOW AND WIND SHEAR
C
C FOR CONVENIENCE, THESE EFFECTS HAVE BEEN INCLUDED IN THE ROTOR MODEL.
C
C EXPANDED VERSION OF VARIABLES FOR EACH BLADE.
C THE 2 BLADES ARE DESIGNATED "A" AND "B"
C
      QI(1)A+2 : 2.+2 : 1.+2


      QI(1)B+2 : 2.+2 : 1.+2


      QDI(1)A+2 : 1.+2 : 1./OR+2


      QDI(1)B+2 : 1.+2 : 1./OR+2


      QDDI(1)A+2 : .2+2 : 1./RAR+2


      QDDI(1)B+2 : .2+2 : 1./RAR+2
```

```
C
C  CONSTANTS APPEARING IN THE SYSTEM EQUATIONS
C
C  JUNE 1985
C
C  1. THE SIMULATION MODEL IS SET UP FOR THE FOLLOWING MAXIMUMS:
C
C        BLADES ..................................(NB)   2
C        RADIAL STATIONS PER BLADE ...............(NR)  11
C        CONTROL SYSTEM STATE VECTOR ELEMENTS ....(NC)   3
C        ELASTIC MODES PER BLADE .................(NM)   3
C        GIMBAL DEGREES OF FREEDOM ...............(NG)   1
C        POWER TRAIN DEGREES OF FREEDOM ..........(NP)   2
C
C  2. THE EXISTING PREPROCESSOR PROGRAMS REQUIRE THAT ALL CONSTANTS USED
C     FOR SCALING AND NORMALIZATION BE DEFINED IN THIS FILE, IRRESPCTIVE
C     OF WHETHER THEY APPEAR EXPLICITLY IN THE SYSTEM EQUATIONS OR NOT.
C
C  3. EVERY CONSTANT LISTED IN THIS FILE MUST APPEAR IN THE DSS2 OUTPUT
C     FILES "DSS2.SNG" OR "DSS2.TBL".
C
C  4. SINCE THE PREPROCESSOR PROGRAMS LIMIT THE RANGE OF ALL CONSTANTS
C     TO +/- 1, THE CONSTANTS "NB" AND "NR" WHICH ARE USED AS PROGRAM
C     LOOPING PARAMETERS, ARE EXPLICITLY DEFINED IN THE SYSTEM EQUATIONS.
C
C-----------------------------------------------------------------------
C
C CONSTANTS USED FOR NORMALIZATION AND SCALING
C
     C34 : 1. : 1.
     BLADE CHORD AT 3/4 RADIUS POINT (FEET).

     FR : 1. : 1.
     REFERENCE FORCE USED FOR NORMALIZATION.

     MR : 1. : 1.
     REFERENCE MOMENT = R*FR

     NB : 1. : 1.
     NUMBER OF BLADES IN THE ROTOR

     OR : 1. : 1.
     REFERENCE ROTATIONAL VELOCITY = ZO (RAD/SEC)

     R : 1. : 1.
     ROTOR RADIUS (FEET)

     RAR : 1. : 1.
     REFERENCE ROTATIONAL ACCELERATION = ZO**2

     VR : 1. : 1.
     REFERENCE TRANSLATIONAL VELOCITY = R*ZO (FT/SEC)

     XAR : 1. : 1.
     REFERENCE TRANSLATIONAL ACCELERATION = R*ZO**2
```

```
       ZO : 1. : 1.
       NOMINAL ROTOR SPIN RATE (RAD/SEC)

C
C  ROTOR MODEL   :  1.1 THROUGH 1.17
C
       CYB : 1. : 1.
       COSINE (YB); YB = 2*PI/NB

       CYC : 1. : 1.
       COSINE (YC); YC = ANGLE TO ROTATE ROTOR AXES TO TEETER AXES

       CYR2 : 1. : 1.
       COSINE (YR2); YR2 = ROTOR SHAFT ANGLE WITH RESPECT TO VERTICAL

       DT : 1. : OR
       NUMERICAL INTEGRATION ON TIME-STEP SIZE, SECONDS.

       DTQ : 5. : OR
       TIME-STEP SIZE, USED IN CORRECTING MODAL QUANTITIES.

       ESQQ11,ESQQ22,ESQQ33 : .1,.05,.01 : 1./RAR+2
       ELEMENTS IN MATRIX OF MODAL FREQUENCIES (RAD/SEC)**2

       FO : .5 : 1.
       BLADE QUIESCENT FEATHERING CONTROL ANGLE.

       FO1,FO2,FO3 : .1,1.,.1 : 1./FR+2
       AERODYNAMIC & INERTIAL FORCES IN THE QUIESCENT STATE

       FQ1(1)+2,FQ2(1)+2,FQ3(1)+2 : .5+2,.5,.125+1,.05+2 : 1./FR+8
       OPERATOR IN SHANK FORCE FORMULA.

       FQD1(1)+2,FQD2(1)+2,FQD3(1)+2 : 1.+2,1.+2,.1+2 : OR/FR+8
       OPERATOR IN SHANK FORCE FORMULA.

       FZ1,FZ2,FZ3 : .1,2.,.2 : 1./FR+2
       AERODYNAMIC FORCES - ROTOR SPEED VARIATIONS

       FZQ1(1)+2,FZQ2(1)+2,FZQ3(1)+2 : 1.+2,1.,.25+1,.1+2 : 1./FR+8
       OPERATOR IN SHANK FORCE FORMULA.

       FZQD1(1)+2,FZQD2(1)+2,FZQD3(1)+2 : 1.+2,1.+2,.1+2 : OR/FR+8
       OPERATOR IN SHANK FORCE FORMULA

       GBO(1)+2 : .5+2 : 1./RAR+2
       CONSTANT COLUMN IN GENERALIZED FORCING FUNCTION FORMULA.

       GQA,GQZA : 1.+1 : RAR/1.+1
       CONSTANT IN GIMBAL SERVO INTEGRATION FORMULA

       GQQ,GQV,GQZV : 1.+2 : 1.,OR/1.+1
       CONSTANT IN GIMBAL SERVO INTEGRATION FORMULA
```

GV(1)1+2,GV(1)2+2,GV(1)3+2 : .25+8 : 1./OR+8
CONSTANT OPERATOR IN GENERALIZED FORCING FUNCTION FORMULA

GVA,GVZA : 1.+1 : OR/1.+1
CONSTANT IN GIMBAL SERVO INTEGRATION FORMULA

GVV,GVQ,GVZQ : 1.+2 : 1.,1./OR+1
CONSTANT IN GIMBAL SERVO INTEGRATION FORMULA

GZQ(1)1+2,GZQ(1)2+2,GZQ(1)3+2 : .25+8 : 1./RAR+8
CONSTANT OPERATOR IN GENERALIZED FORCING FUNCTION FORMULA

GZV(1)1+2,GZV(1)2+2,GZV(1)3+2 : .25+8 : 1./OR+8
CONSTANT OPERATOR IN GENERALIZED FORCING FUNCTION FORMULA

KGE : 20. : MR/RAR
CONSTANT IN GIMBAL SERVO INTEGRATION FORMULA

KGQC : 1. : NB/RAR
CONSTANT IN GIMBAL SERVO INTEGRATION FORMULA

KGQS : 1. : NB/RAR
CONSTANT IN GIMBAL SERVO INTEGRATION FORMULA

KGVC : 1. : NB/OR
CONSTANT IN GIMBAL SERVO INTEGRATION FORMULA

KGVS : 1. : NB/OR
CONSTANT IN GIMBAL SERVO INTEGRATION FORMULA

KQG11,KQG22,KQG33 : 8.+2 : RAR+2
CONSTANT IN BLADE NORMAL COORDINATE INTEGRATION ALGORITHM

KQQ11,KQQ22,KQQ33 : 1.+2 : 1.+2
CONSTANTS IN BLADE NORMAL COORDINATE INTEGRATION ALGORITHM

KQV11,KQV22,KQV33 : 2.+2 : OR+2
CONSTANT IN BLADE NORMAL COORDINATE INTEGRATION ALGORITHM

KQZG11,KQZG22,KQZG33 : 8.+2 : RAR+2
CONSTANT IN BLADE NORMAL COORDINATE INTEGRATION ALGORITHM

KQZQ11,KQZQ22,KQZQ33 : 1.+2 : 1.+2
CONSTANT IN BLADE NORMAL COORDINATE INTEGRATION ALGORITHM

KQZV11,KQZV22,KQZV33 : 2.+2 : OR+2
CONSTANT IN BLADE NORMAL COORDINATE INTEGRATION ALGORITHM

KVG11,KVG22,KVG33 : 4.+2 : OR+2
CONSTANT IN BLADE NORMAL COORDINATE INTEGRATION ALGORITHM

KVQ11,KVQ22,KVQ33 : .5+2 : 1./OR+2
CONSTANT IN BLADE NORMAL COORDINATE INTEGRATION ALGORITHM

KVV11,KVV22,KVV33 : 1.+2 : 1.+2

```
         CONSTANTS IN NORMAL COORDINATE INTEGRATION ALGORITHM

         KVZG11,KVZG22,KVZG33 : 4.+2 : OR+2
         CONSTANT IN BLADE NORMAL COORDINATE INTEGRATION ALGORITHM

         KVZQ11,KVZQ22,KVZQ33 : .5+2 : 1./OR+2
         CONSTANT IN BLADE NORMAL COORDINATE INTEGRATION ALGORITHM

         KVZV11,KVZV22,KVZV33 : 1.+2 : 1+2
         CONSTANT IN BLADE NORMAL COORDINATE INTEGRATION ALGORITHM

         MO1,MO2,MO3 : .25,.25,1. : 1./MR+2
         AERODYNAMIC MOMENTS IN THE QUIESCENT STATE

         MQ1(1)+2,MQ2(1)+2,MQ3(1)+2 : .125+2,.125+2,.5,.25+1 : 1./MR+8
         OPERATOR IN SHANK FORCE FORMULA

         MQD1(1)+2,MQD2(1)+2,MQD3(1)+2 : .25+2,.25+2,1.+2 : OR/MR+8
         OPERATOR IN SHANK FORCE FORMULA

         MZ1,MZ2,MZ3 : .5,.5,2. : 1./MR+2
         AERODYNAMIC MOMENTS - ROTOR SPEED VARIATIONS

         MZQ1(1)+2,MZQ2(1)+2,MZQ3(1)+2 : .25+2,.25+2,1.,.5+1 : 1./MR+8
         OPERATOR IN SHANK FORCE FORMULA

         MZQD1(1)+2,MZQD2(1)+2,MZQD3(1)+2 : .25+2,.25+2,1.+2 : OR/MR+8
         OPERATOR IN SHANK FORCE FORMULA

         PR : 1. : 1.
         AIR PRESSURE RATIO, P/PO; (ACTUAL/STANDARD SEA LEVEL)

         QFACT : .01 : NB*R*C34/FR
         FACTOR; AERODYNAMIC PRESSURE = 1481

         RNB : 1. : 1.
         THE RATIO 1./NB

         RVC : 2. : VR
         RECIPROCAL OF THE SPEED OF SOUND

         SYB : 1. : 1.
         SINE (YB); YB = 2*PI/NB

         SYC : 1. : 1.
         SINE (YC); YC = ANGLE TO ROTATE ROTOR AXES TO TEETER AXES

         SYR2 : 1. : 1.
         SINE (YR2); YR2 = ROTOR SHAFT ANGLE WITH RESPECT TO VERTICAL

         ZU : 10. : 1./R
         ROTOR UNDER- OR SIDE-SLING.

C
C TOWER SHADOW AND WIND SHEAR
```

```
C
C FOR CONVENIENCE THESE EFFECTS HAVE BEEN INCLUDED IN THE ROTOR MODEL.
C NORMALLY THEY ARE PART OF THE "AIR MASS" SUBSYSTEM.
C
     SHADOW : 1. : 1.
     AIR VELOCITY REDUCTION DUE TO TOWER SHADOW :  (.28 => 28 %).

     SHDSEC : 1. : 1
     COSINE OF HALF THE ANGLE DEFINING THE TOWER SHADOW SECTOR.

     SHEAR : 1. : 1.
     WIND SHEAR STRENGTH EXPRESSED AS A FRACTION. (.15 => 15 %)
```

```
C
C SUPPORT SYSTEM  :  2.0
C
      DS11,DS22,DS33 : .5,.5,.25 : 1./OR+2
      DAMPING (DIAGONAL) MATRIX-SUPPORT SYSTEM MODAL EQUATIONS.

      DTS : 10. : OR
      NUMERICAL INTEGRATION DT :  SUPPORT SYSTEM

      GSF11,GSF13,GSF22,GSF32 : .5,2.5,.5,.5 : FR/RAR+3
      PARTICIPATION FACTOR ON FORCE-SUPPORT SYSTEM

      GSM12,GSM23,GSM31 : 2.,.5,1. : MR/RAR+2
      PARTICIPATION FACTORS ON MOMENT-SUPPORT SYSTEM

      SV11,SV22,SV23,SV31 : 20.,20.,20.,20. : 1./R+3
      TRANSLATIONAL VELOCITY MODAL MATRIX-SUPPORT SYSTEM

      SW13,SW21,SW32 : 5.,5.,10. : 1.+2
      ROTATIONAL VELOCITY MODAL MATRIX-SUPPORT SYSTEM.

      W2S11,W2S22,W2S33 : .05,.05,.025 : 1./RAR+2
      FREQUENCY SQUARED(DIAGONAL)MATRIX-SUPPORT SYSTEM MODAL EQUATIONS.

C
C CONTROLS  :  3.0
C
      AA21,AA22 : .01,.04 : 1./OR+1
      CHARACTERISTIC MATRIX IN CONTROL SYSTEM STATE EQUATION.

      AD03 : .5 : 1./OR
      CONTROL SYSTEM PARTICIPATION FACTOR

      AG3 : 1. : 1.
      CONTROL SYSTEM PARTICIPATION FACTOR

      AZ1,AZ2 : 1.,.01 : 1.+1
      CONTROL SYSTEM PARTICIPATION FACTORS

      CA12 : 1. : 1.
      CONTROL SYSTEM RESULT OPERATOR,A TO C.

      DTC1,DTC2,DTC3 : 1.,12.5,10. : OR+2
      NUMERICAL INTEGRATION DT :  CONTROLS

      LA13 : .1 : 1./MR
      CONTROL SYSTEM RESULT,A TO ALTERNATOR CONTROL

      LG : 1. : OR/MR
      CONTROL SYSTEM RESULT OPERATOR

      LPR : 1. : 1./MR
      CONTROL SYSTEM RESULT OPERATOR

      PRQ : .5 : 1.
      REQUESTED POWER RATIO (REQUESTED POWER/RATED POWER)

      SRQ : .5 : 1.
      REQUESTED SPEED RATIO = ZREQ/ZO

C
C POWER TRAIN  :  4.0
C
      BP11,BP12,BP21,BP22 : 5.,.5,1.,.1 : 1./OR+3
      POWER TRAIN DAMPING MATRIX

      DTP1,DTP2 : .1,10. : OR+1
      NUMERICAL INTEGRATION DT :  POWER TRAIN.

      KP12,KP22 : .1,.02 : 1./RAR+1
      POWER TRAIN STIFFNESS MATRIX.

      PL2 : .125 : MR/RAR
      CONTROL SYSTEM INFLUENCE FACTOR-POWER TRAIN MODAL.

      PM1,PM2 : 2.5,.5 : MR/RAR+1
      ROTOR MOMENT INFLUENCE FACTOR-POWER TRAIN MODAL.

      PZD1,PZD2 : .5,.1 : 1.+1
      ROTOR ACCELERATION INFLUENCE FACTOR-POWER TRAIN MODEL.

C
C AIR MASS  :  5.0
C
      DTG : 16. : OR
      GUST FILTER NUMERICAL INTEGRATION STEP SIZE

      GCEV : .0625 : 1.
      GAIN IN RETARDATION VELOCITY SERVO.

      KAR : 625. : FR/VR*VR
      GLAUERT CONSTANT = 1./(2.*PI*R*R*RHOO); RHOO = STANDARD AIR DENSITY

      RHOR : .5 : 1.
      AIR DENSITY RATIO = RHOO/RHO; (STANDARD SEA LEVEL / ACTUAL)

      VGU11,VGU22,VGU33 : 2.5+2 : 1./R+2
      WINDGUST TRANSLATIONAL VELOCITY COEFFICIENT MATRIX

      WGG11,WGG22,WGG33,WGG44,WGG55,WGG66 : .0625+5 : 1./RAR+5
      WINDGUST FILTER ROTATIONAL FREQUENCIES (RAD/SEC).

      WGU11,WGU22,WGU33 : 2.5+2 : 1.+2
      WINDGUST ROTATIONAL VELOCITY COEFFICIENT MATRIX

      ZTG11,ZTG22,ZTG33,ZTG44,ZTG55,ZTG66 : 1.+5 : 1.+5
      DAMPING COEFFICIENTS IN WINDGUST FILTERS.

C
C TOWER SHADOW AND WIND SHEAR
```

```
C
C FOR CONVENIENCE THESE EFFECTS HAVE BEEN INCLUDED IN THE ROTOR MODEL.
C
C EXPANDED VERSION OF CONSTANTS AT EACH RADIAL STATION.
C THE 11 RADIAL STATIONS ARE DESIGNATED "A" THROUGH "K".
C
      BOO(1)A+2 : 1.,5.+1 : 1./R+2

      BOO(1)B+2 : 1.,5.+1 : 1./R+2

      BOO(1)C+2 : 1.,5.+1 : 1./R+2

      BOO(1)D+2 : 1.,5.+1 : 1./R+2

      BOO(1)E+2 : 1.,5.+1 : 1./R+2

      BOO(1)F+2 : 1.,5.+1 : 1./R+2

      BOO(1)G+2 : 1.,5.+1 : 1./R+2

      BOO(1)H+2 : 1.,5.+1 : 1./R+2

      BOO(1)I+2 : 1.,5.+1 : 1./R+2

      BOO(1)J+2 : 1.,5.+1 : 1./R+2

      BOO(1)K+2 : 1.,5.+1 : 1./R+2

      CBA : .5 : 1./C34

      CBB : .5 : 1./C34

      CBC : .5 : 1./C34

      CBD : .5 : 1./C34

      CBE : .5 : 1./C34

      CBF : .5 : 1./C34
```

```
CBG : .5 : 1./C34

CBH : .5 : 1./C34

CBI : .5 : 1./C34

CBJ : .5 : 1./C34

CBK : .5 : 1./C34

DA(1)1A+2,DA(1)2A+2,DA(1)3A+2 : 1.+8 : 1.+8

DA(1)1B+2,DA(1)2B+2,DA(1)3B+2 : 1.+8 : 1.+8

DA(1)1C+2,DA(1)2C+2,DA(1)3C+2 : 1.+8 : 1.+8

DA(1)1D+2,DA(1)2D+2,DA(1)3D+2 : 1.+8 : 1.+8

DA(1)1E+2,DA(1)2E+2,DA(1)3E+2 : 1.+8 : 1.+8

DA(1)1F+2,DA(1)2F+2,DA(1)3F+2 : 1.+8 : 1.+8

DA(1)1G+2,DA(1)2G+2,DA(1)3G+2 : 1.+8 : 1.+8

DA(1)1H+2,DA(1)2H+2,DA(1)3H+2 : 1.+8 : 1.+8

DA(1)1I+2,DA(1)2I+2,DA(1)3I+2 : 1.+8 : 1.+8

DA(1)1J+2,DA(1)2J+2,DA(1)3J+2 : 1.+8 : 1.+8

DA(1)1K+2,DA(1)2K+2,DA(1)3K+2 : 1.+8 : 1.+8

DB(1)1A+2,DB(1)2A+2,DB(1)3A+2 : 5.,2.5+1,5.,2.5+1,5.,2.5+1 : 1./R+8

DB(1)1B+2,DB(1)2B+2,DB(1)3B+2 : 5.,2.5+1,5.,2.5+1,5.,2.5+1 : 1./R+8
DB(1)1C+2,DB(1)2C+2,DB(1)3C+2 : 5.,2.5+1,5.,2.5+1,5.,2.5+1 : 1./R+8

DB(1)1D+2,DB(1)2D+2,DB(1)3D+2 : 5.,2.5+1,5.,2.5+1,5.,2.5+1 : 1./R+8

DB(1)1E+2,DB(1)2E+2,DB(1)3E+2 : 5.,2.5+1,5.,2.5+1,5.,2.5+1 : 1./R+8

DB(1)1F+2,DB(1)2F+2,DB(1)3F+2 : 5.,2.5+1,5.,2.5+1,5.,2.5+1 : 1./R+8

DB(1)1G+2,DB(1)2G+2,DB(1)3G+2 : 5.,2.5+1,5.,2.5+1,5.,2.5+1 : 1./R+8

DB(1)1H+2,DB(1)2H+2,DB(1)3H+2 : 5.,2.5+1,5.,2.5+1,5.,2.5+1 : 1./R+8

DB(1)1I+2,DB(1)2I+2,DB(1)3I+2 : 5.,2.5+1,5.,2.5+1,5.,2.5+1 : 1./R+8

DB(1)1J+2,DB(1)2J+2,DB(1)3J+2 : 5.,2.5+1,5.,2.5+1,5.,2.5+1 : 1./R+8

DB(1)1K+2,DB(1)2K+2,DB(1)3K+2 : 5.,2.5+1,5.,2.5+1,5.,2.5+1 : 1./R+8

DSA : 10. : 1./R

DSB : 10. : 1./R

DSC : 10. : 1./R

DSD : 10. : 1./R

DSE : 10. : 1./R

DSF : 10. : 1./R

DSG : 10. : 1./R

DSH : 10. : 1./R

DSI : 10. : 1./R
```

```
DSJ : 10. : 1./R

DSK : 10. : 1./R

FAO(1)A+2 : .5,.25,.1 : NB*R/FR+2

FAO(1)B+2 : .5,.25,.1 : NB*R/FR+2

FAO(1)C+2 : .5,.25,.1 : NB*R/FR+2

FAO(1)D+2 : .5,.25,.1 : NB*R/FR+2

FAO(1)E+2 : .5,.25,.1 : NB*R/FR+2

FAO(1)F+2 : .5,.25,.1 : NB*R/FR+2

FAO(1)G+2 : .5,.25,.1 : NB*R/FR+2

FAO(1)H+2 : .5,.25,.1 : NB*R/FR+2

FAO(1)I+2 : .5,.25,.1 : NB*R/FR+2

FAO(1)J+2 : .5,.25,.1 : NB*R/FR+2

FAO(1)K+2 : .5,.25,.1 : NB*R/FR+2

FP(1)1A+2,FP(1)2A+2,FP(1)3A+2 : .2+2,.4+2,1.,1.,.5 : 1.+8

FP(1)1B+2,FP(1)2B+2,FP(1)3B+2 : .2+2,.4+2,1.,1.,.5 : 1.+8

FP(1)1C+2,FP(1)2C+2,FP(1)3C+2 : .2+2,.4+2,1.,1.,.5 : 1.+8

FP(1)1D+2,FP(1)2D+2,FP(1)3D+2 : .2+2,.4+2,1.,1.,.5 : 1.+8

FP(1)1E+2,FP(1)2E+2,FP(1)3E+2 : .2+2,.4+2,1.,1.,.5 : 1.+8

FP(1)1F+2,FP(1)2F+2,FP(1)3F+2 : .2+2,.4+2,1.,1.,.5 : 1.+8
FP(1)1G+2,FP(1)2G+2,FP(1)3G+2 : .2+2,.4+2,1.,1.,.5 : 1.+8

FP(1)1H+2,FP(1)2H+2,FP(1)3H+2 : .2+2,.4+2,1.,1.,.5 : 1.+8

FP(1)1I+2,FP(1)2I+2,FP(1)3I+2 : .2+2,.4+2,1.,1.,.5 : 1.+8

FP(1)1J+2,FP(1)2J+2,FP(1)3J+2 : .2+2,.4+2,1.,1.,.5 : 1.+8

FP(1)1K+2,FP(1)2K+2,FP(1)3K+2 : .2+2,.4+2,1.,1.,.5 : 1.+8

GP(1)1A+2,GP(1)2A+2,GP(1)3A+2 : .05+2,.1+2,.25+2 : FR/NB*RAR+8

GP(1)4A+2 : 1.,1.,.8 : FR/NB*RAR+1,FR*C34/NB*R*RAR

GP(1)1B+2,GP(1)2B+2,GP(1)3B+2 : .05+2,.1+2,.25+2 : FR/NB*RAR+8

GP(1)4B+2 : 1.,1.,.8 : FR/NB*RAR+1,FR*C34/NB*R*RAR

GP(1)1C+2,GP(1)2C+2,GP(1)3C+2 : .05+2,.1+2,.25+2 : FR/NB*RAR+8

GP(1)4C+2 : 1.,1.,.8 : FR/NB*RAR+1,FR*C34/NB*R*RAR

GP(1)1D+2,GP(1)2D+2,GP(1)3D+2 : .05+2,.1+2,.25+2 : FR/NB*RAR+8

GP(1)4D+2 : 1.,1.,.8 : FR/NB*RAR+1,FR*C34/NB*R*RAR

GP(1)1E+2,GP(1)2E+2,GP(1)3E+2 : .05+2,.1+2,.25+2 : FR/NB*RAR+8

GP(1)4E+2 : 1.,1.,.8 : FR/NB*RAR+1,FR*C34/NB*R*RAR

GP(1)1F+2,GP(1)2F+2,GP(1)3F+2 : .05+2,.1+2,.25+2 : FR/NB*RAR+8

GP(1)4F+2 : 1.,1.,.8 : FR/NB*RAR+1,FR*C34/NB*R*RAR

GP(1)1G+2,GP(1)2G+2,GP(1)3G+2 : .05+2,.1+2,.25+2 : FR/NB*RAR+8
GP(1)4G+2 : 1.,1.,.8 : FR/NB*RAR+1,FR*C34/NB*R*RAR

GP(1)1H+2,GP(1)2H+2,GP(1)3H+2 : .05+2,.1+2,.25+2 : FR/NB*RAR+8

GP(1)4H+2 : 1.,1.,.8 : FR/NB*RAR+1,FR*C34/NB*R*RAR

GP(1)1I+2,GP(1)2I+2,GP(1)3I+2 : .05+2,.1+2,.25+2 : FR/NB*RAR+8

GP(1)4I+2 : 1.,1.,.8 : FR/NB*RAR+1,FR*C34/NB*R*RAR

GP(1)1J+2,GP(1)2J+2,GP(1)3J+2 : .05+2,.1+2,.25+2 : FR/NB*RAR+8

GP(1)4J+2 : 1.,1.,.8 : FR/NB*RAR+1,FR*C34/NB*R*RAR

GP(1)1K+2,GP(1)2K+2,GP(1)3K+2 : .05+2,.1+2,.25+2 : FR/NB*RAR+8

GP(1)4K+2 : 1.,1.,.8 : FR/NB*RAR+1,FR*C34/NB*R*RAR

MBA : .025 : NB*VR*VR/FR

MBB : .025 : NB*VR*VR/FR

MBC : .025 : NB*VR*VR/FR

MBD : .025 : NB*VR*VR/FR

MBE : .025 : NB*VR*VR/FR

MBF : .025 : NB*VR*VR/FR

MBG : .025 : NB*VR*VR/FR

MBH : .025 : NB*VR*VR/FR

MBI : .025 : NB*VR*VR/FR
```

```
MBJ : .025 : NB*VR*VR/FR

MBK : .025 : NB*VR*VR/FR

MP(1)1A+2,MP(1)2A+2,MP(1)3A+2 : .2,.5+1,.4,1.+1,1.,1.25,2.5 : 1./R+8

MP(1)1B+2,MP(1)2B+2,MP(1)3B+2 : .2,.5+1,.4,1.+1,1.,1.25,2.5 : 1./R+8

MP(1)1C+2,MP(1)2C+2,MP(1)3C+2 : .2,.5+1,.4,1.+1,1.,1.25,2.5 : 1./R+8

MP(1)1D+2,MP(1)2D+2,MP(1)3D+2 : .2,.5+1,.4,1.+1,1.,1.25,2.5 : 1./R+8

MP(1)1E+2,MP(1)2E+2,MP(1)3E+2 : .2,.5+1,.4,1.+1,1.,1.25,2.5 : 1./R+8

MP(1)1F+2,MP(1)2F+2,MP(1)3F+2 : .2,.5+1,.4,1.+1,1.,1.25,2.5 : 1./R+8

MP(1)1G+2,MP(1)2G+2,MP(1)3G+2 : .2,.5+1,.4,1.+1,1.,1.25,2.5 : 1./R+8

MP(1)1H+2,MP(1)2H+2,MP(1)3H+2 : .2,.5+1,.4,1.+1,1.,1.25,2.5 : 1./R+8

MP(1)1I+2,MP(1)2I+2,MP(1)3I+2 : .2,.5+1,.4,1.+1,1.,1.25,2.5 : 1./R+8

MP(1)1J+2,MP(1)2J+2,MP(1)3J+2 : .2,.5+1,.4,1.+1,1.,1.25,2.5 : 1./R+8

MP(1)1K+2,MP(1)2K+2,MP(1)3K+2 : .2,.5+1,.4,1.+1,1.,1.25,2.5 : 1./R+8

RFSA : 1. : 1.

RFSB : 1. : 1.

RFSC : 1. : 1.

RFSD : 1. : 1.

RFSE : 1. : 1.

RFSF : 1. : 1.
```

```
RFSG : 1. : 1.

RFSH : 1. : 1.

RFSI : 1. : 1.

RFSJ : 1. : 1.

RFSK : 1. : 1.

ROO(1)1A+2,ROO(1)2A+2,ROO(1)3A+2 : 1.,2.+2,1.+1,2.,1.+1 : 1.+8

ROO(1)1B+2,ROO(1)2B+2,ROO(1)3B+2 : 1.,2.+2,1.+1,2.,1.+1 : 1.+8

ROO(1)1C+2,ROO(1)2C+2,ROO(1)3C+2 : 1.,2.+2,1.+1,2.,1.+1 : 1.+8

ROO(1)1D+2,ROO(1)2D+2,ROO(1)3D+2 : 1.,2.+2,1.+1,2.,1.+1 : 1.+8

ROO(1)1E+2,ROO(1)2E+2,ROO(1)3E+2 : 1.,2.+2,1.+1,2.,1.+1 : 1.+8

ROO(1)1F+2,ROO(1)2F+2,ROO(1)3F+2 : 1.,2.+2,1.+1,2.,1.+1 : 1.+8

ROO(1)1G+2,ROO(1)2G+2,ROO(1)3G+2 : 1.,2.+2,1.+1,2.,1.+1 : 1.+8

ROO(1)1H+2,ROO(1)2H+2,ROO(1)3H+2 : 1.,2.+2,1.+1,2.,1.+1 : 1.+8

ROO(1)1I+2,ROO(1)2I+2,ROO(1)3I+2 : 1.,2.+2,1.+1,2.,1.+1 : 1.+8

ROO(1)1J+2,ROO(1)2J+2,ROO(1)3J+2 : 1.,2.+2,1.+1,2.,1.+1 : 1.+8

ROO(1)1K+2,ROO(1)2K+2,ROO(1)3K+2 : 1.,2.+2,1.+1,2.,1.+1 : 1.+8

SA : 1. : 1./R

SB : 1. : 1./R
```

```
SC : 1. : 1./R

SD : 1. : 1./R

SE : 1. : 1./R

SF : 1. : 1./R

SG : 1. : 1./R

SH : 1. : 1./R

SI : 1. : 1./R

SJ : 1. : 1./R

SK : 1. : 1./R
```

APPENDIX G

PHYSICAL EQUATIONS OF THE WIND TURBINE SIMULATION MODEL

Presented in this Appendix are the physical equations which constitute the simulation model of a wind turbine. These equations have to be scaled and normalized for use in WEST-3.

```
C
C**********************************************************************
C
C WEST-3 PHYSICAL EQUATIONS: JUNE 1985
C
C**********************************************************************
C
C
C INITIALIZATIONS
C
        NB = 2
        NR = 11
C
        Z = ZINT
        ZD= 0
C
        CY = 1.
        SY = 0
        CDY= 1.
C
        VH1= VBH1
        VH2= VBH2
        VH3= VBH3
        WH1= WBH1
        WH2= WBH2
        WH3= WBH3
C
        VDH1= VDBH1
        VDH2= VDBH2
        VDH3= VDBH3
        WDH1= WDBH1
        WDH2= WDBH2
        WDH3= WDBH3
C
        VWT1  = VW1
        VWT2  = VW2
        VWT3  = VW3
        WWT1  = WW1
        WWT2  = WW2
        WWT3  = WW3
        VRW   = 0
C
        S1  = 0
        S2  = 0
        S3  = 0
        SD1= 0
        SD2= 0
        SD3= 0
C
        C1  = 0
        A1  = 0
        A2  = 0
        A3  = 0
        ZA  = 0
C
```

```
          PD1= Z
          P2 = 0
          PD2= 0
C
C SET ALL MODAL QUANTITIES TO ZERO
C
          CALL LOADQS (0)
C
C
C***************** BEGINING OF INFINITE LOOP  *************************
C
 9000 CONTINUE
C
C
C AEROELASTIC ROTOR MODEL
C
C ROTOR SPEED VARIABLES
C
          ZP = Z/Z0 - 1.
          ZZ = (Z/Z0)*(Z/Z0) - 1.
          ZSQ = Z * Z
C
C 1.2 HUB TO SHANK TRANSFORMATIONS
C
C     MANY OF THE TRANSFORMATIONS ARE COMMON TO ALL THE
C     BLADES, AND ARE PROCESSED OUTSIDE THE BLADE LOOP.
C
C GEOMETRY
C
          SYT = SYC*CY - CYC*SY
          CYT = CYC*CY + SYC*SY
C
C INERTIAL VELOCITY
C
          WTUMZ1 = CYT*WH1 + SYT*WH2
          WTUMZ2 =-SYT*WH1 + CYT*WH2
          WTUMZ3 = WH3
          VTU1 = CYT*VH1 + SYT*VH2 + ZU*WTUMZ2
          VTU2 =-SYT*VH1 + CYT*VH2 - ZU*WTUMZ1
          VTU3 = VH3
C
          WCMZ1 = WTUMZ1 - WTUMZ3*QG1 + Z*QG1
          WCMZ2 = WTUMZ2 + QDG1
          WCMZ3 = WTUMZ3 + WTUMZ1*QG1
          VC1 = VTU1 - VTU3*QG1 - ZU*WCMZ2
          VC2 = VTU2 + ZU*WCMZ1
          VC3 = VTU3 + VTU1*QG1
C
C AERO VELOCITY
C
          WAH1 = WH1 + WWT1*CYR2 - WWT3*SYR2
          WAH2 = WH2 + WWT2
          WAH3 = WH3 + WWT1*SYR2 + WWT3*CYR2
          VAH1 = VH1 + VWT1*CYR2 - VWT3*SYR2
          VAH2 = VH2 + VWT2
```

```
      VAH3 = VH3 + VWT1*SYR2 + VWT3*CYR2
C
      WATU1 = CYT*WAH1 + SYT*WAH2
      WATU2 =-SYT*WAH1 + CYT*WAH2
      WATU3 = WAH3 - Z
      VATU1 = CYT*VAH1 + SYT*VAH2 + ZU*WATU2
      VATU2 =-SYT*VAH1 + CYT*VAH2 - ZU*WATU1
      VATU3 = VAH3
C
      WAC1 = WATU1 - WATU3*QG1
      WAC2 = WATU2 + QDG1
      WAC3 = WATU3 + WATU1*QG1
      VAC1 = VATU1 - VATU3*QG1 - ZU*WAC2
      VAC2 = VATU2 + ZU*WAC1
      VAC3 = VATU3 + VATU1*QG1
C
C TIME DERIVATIVE OF INERTIAL VELOCITY.
C
      DUM1  = VDH1 - Z*VH2
      DUM2  = VDH2 + Z*VH1
      TEMP1 = WDH1 - Z*WH2
      TEMP2 = WDH2 + Z*WH1
C
      WDTU1 = CYT*TEMP1 + SYT*TEMP2
      WDTU2 =-SYT*TEMP1 + CYT*TEMP2
      WDTU3 = WDH3-ZD
      VDTU1 = CYT*DUM1 + SYT*DUM2 + ZU*WDTU2
      VDTU2 =-SYT*DUM1 + CYT*DUM2 - ZU*WDTU1
C
      DUM3 = WDTU1 - WTUMZ3*QDG1 + Z*QDG1
      DUM4 = WDTU3 + WTUMZ1*QDG1
      TEMP3= VDTU1 - VTU3*QDG1
      TEMP4= VDH3  + VTU1*QDG1
C
      WDC1 = DUM3  - DUM4*QG1
      WDC2 = WDTU2 + QDDG1
      WDC3 = DUM4  + DUM3*QG1
      VDC1 = TEMP3 - TEMP4*QG1 - ZU*WDC2
      VDC2 = VDTU2 + ZU*WDC1
      VDC3 = TEMP4 + TEMP3*QG1
C
C  INITIALIZATIONS FOR BLADE LOOP
C
      I = 0
C
      SYBI =-SYB
      CYBI = CYB
C
C     MULTIBLADE COORDINATE AND LOADS.
C
      QS1=0
      QC1=0
      Q01=0
      QDS1=0
      QDC1=0
      QD01=0
C
      QS2=0
      QC2=0
      Q02=0
      QDS2=0
      QDC2=0
      QD02=0
C
      QS3=0
      QC3=0
      Q03=0
      QDS3=0
      QDC3=0
      QD03=0
C
      FGR1=0
      FGR2=0
      FGR3=0
C
      MGR1=0
      MGR2=0
      MGR3=0
C
C*****************   BEGINING OF BLADE LOOP   ************************
C
C 1.1  BLADE NUMBER SELECT
C
 9001  CONTINUE
C
      I = I + 1
C
      SYBISV = SYBI
      SYBI = SYB*CYBI + CYB*SYBI
      CYBI = CYB*CYBI - SYB*SYBISV
C
C LOAD QI,QDI AND QDDI FOR BLADE I FROM
C SPACE CONTAINING Q, QD AND QDD FOR ALL BLADES.
C
      CALL LOADQS(I)
C
C
C 1.2 HUB TO SHANK TRANSFORMATIONS
C
C     THESE TRANSFORMATIONS DEPEND ON THE BLADE AZIMUTHAL POSITION
C
C GEOMETRY
C
      CYI = CY*CYBI - SY*SYBI
      SYI = SY*CYBI + CY*SYBI
C
      SYCBI = SYC*CYBI + CYC*SYBI
      CYCBI = CYC*CYBI - SYC*SYBI
C
C BLADE FEATHERING CONTROL ANGLE
```

```
C
      F = C1 - C2*CYI - C3*SYI
C
C AERO VELOCITY
C
      WAS1 = CYCBI*WAC1 - SYCBI*WAC2
      WAS2 = SYCBI*WAC1 + CYCBI*WAC2
      WAS3 = WAC3
      VAS1 = CYCBI*VAC1 - SYCBI*VAC2
      VAS2 = SYCBI*VAC1 + CYCBI*VAC2
      VAS3 = VAC3
C
C INERTIAL VELOCITY
C
      WSMZ1 = CYCBI*WCMZ1 - SYCBI*WCMZ2
      WSMZ2 = SYCBI*WCMZ1 + CYCBI*WCMZ2
      WSMZ3 = WCMZ3
      VS1 = CYCBI*VC1 - SYCBI*VC2
      VS2 = SYCBI*VC1 + CYCBI*VC2
      VS3 = VC3
C
C TIME DERIVATIVE OF INERTIAL VELOCITY.
C
      WDS1 = CYCBI*WDC1 - SYCBI*WDC2
      WDS2 = SYCBI*WDC1 + CYCBI*WDC2
      WDS3 = WDC3
      VDS1 = CYCBI*VDC1 - SYCBI*VDC2
      VDS2 = SYCBI*VDC1 + CYCBI*VDC2
      VDS3 = VDC3
C
C ACCELERATION TERMS FOR USE IN SECTION 1.7
C
C MANY OF THE TERMS NEEDED FOR COMPUTING THE DISTRIBUTED
C INERTIA LOADS AT EACH BLADE RADIAL STATION ARE COMMON TO
C ALL THE STATIONS, AND ARE GENERATED OUTSIDE THE BLADE LOOP
C
      US1 = VDS1 + WSMZ2*VS3 - WSMZ3*VS2 + Z*VS2
      US2 = VDS2 + WSMZ3*VS1 - WSMZ1*VS3 - Z*VS1
      US3 = VDS3 + WSMZ1*VS2 - WSMZ2*VS1
C
C THE PHYSICAL EQUATION FOR THE NEXT STATEMENT IS ... TERM = WSMZ3 - 2 * Z
C
      TERM   = WSMZ3 - 2.*Z
      U11 = -WSMZ3*TERM  - WSMZ2*WSMZ2
      U22 = -WSMZ3*TERM  - WSMZ1*WSMZ1
      U33 = -WSMZ2*WSMZ2 - WSMZ1*WSMZ1
C
      U12 = WSMZ2*WSMZ1
      U13 = WSMZ3*WSMZ1 - Z*WSMZ1
      U23 = WSMZ3*WSMZ2 - Z*WSMZ2
C
C     INITIALIZE FOR RADIAL INTEGRATION OF BLADE,I.
C
      J = 0
C
```

```
        X1=0
        X2=0
        X3=0
        XD1=0
        XD2=0
        XD3=0
        XDD1=0
        XDD2=0
        XDD3=0
C
        GRI1=0
        GRI2=0
        GRI3=0
C
        FRI1=0
        FRI2=0
        FRI3=0
C
        MRI1=0
        MRI2=0
        MRI3=0
C
C*****************   BEGINING OF RADIAL LOOP  ************************
C
C 1.3 RADIAL STATION COUNTER.
C
 9002 CONTINUE
      J = J + 1
C
C 1.4 RADIAL PHYSICAL PARAMETER BLOCKMOVE.  MOVE ALL BLADE PHYSICAL
C     PROPERTIES FROM A COMMON MEMORY REGION TO A SEPARATE REGION
C     APPLICABLE FOR A SINGLE BLADE ELEMENT.
C
C     ENTITIES MOVED (AND THEIR SIZES) ARE-
C
C     ROO(9),RFS,DA(3*NM),DB(3*NM),BOO(3),GP(4*NM),FP(9),MP(9),
C     FAO(3),MB,CB
C
        CALL MOVIT(J)
C
C
C 1.5 BLADE ELEMENT GEOMETRY
C
C MODAL ANGULAR ACCELERATION: BAX WITH RESPECT TO SAX.
C
        EA1   = DA11*QI1   + DA12*QI2   + DA13*QI3
        EDA1  = DA11*QDI1  + DA12*QDI2  + DA13*QDI3
        EDDA1 = DA11*QDDI1 + DA12*QDDI2 + DA13*QDDI3
C
        EA2   = DA21*QI1   + DA22*QI2   + DA23*QI3
        EDA2  = DA21*QDI1  + DA22*QDI2  + DA23*QDI3
        EDDA2 = DA21*QDDI1 + DA22*QDDI2 + DA23*QDDI3
C
        EA3   = DA31*QI1   + DA32*QI2   + DA33*QI3
        EDA3  = DA31*QDI1  + DA32*QDI2  + DA33*QDI3
      EDDA3 = DA31*QDDI1 + DA32*QDDI2 + DA33*QDDI3
C
C LINEAR MODAL TRANSLATIONAL BAX MOVEMENT WITH RESPECT TO SAX.
C
      EB1   = DB11*QI1   + DB12*QI2   + DB13*QI3
      EDB1  = DB11*QDI1  + DB12*QDI2  + DB13*QDI3
      EDDB1 = DB11*QDDI1 + DB12*QDDI2 + DB13*QDDI3
C
      EB2   = DB21*QI1   + DB22*QI2   + DB23*QI3
      EDB2  = DB21*QDI1  + DB22*QDI2  + DB23*QDI3
      EDDB2 = DB21*QDDI1 + DB22*QDDI2 + DB23*QDDI3
C
      EB3   = DB31*QI1   + DB32*QI2   + DB33*QI3
      EDB3  = DB31*QDI1  + DB32*QDI2  + DB33*QDI3
      EDDB3 = DB31*QDDI1 + DB32*QDDI2 + DB33*QDDI3
C
C APPROXIMATIONS FOR SINE AND COSINE OF FEATHERING ANGLE, FS.
C IF "RFS" IS CONSTANT, THESE CAN BE OUTSIDE THE RADIAL LOOP.
C
      FS = RFS*F - RFS*F0
      FSQ = FS*FS
      SFS = FS*(1.-.16667*FSQ*(1.-.05*FSQ))
      CFS = 1. - .5*FSQ*(1.-.08333*FSQ)
C
C TRANSFORMATION FROM SAX TO BAX CONTROL POSITION
C
      R011 = R0011
      R021 = R0021*CFS + R0031*SFS
      R031 =-R0021*SFS + R0031*CFS
      R012 = R0012
      R022 = R0022*CFS + R0032*SFS
      R032 =-R0022*SFS + R0032*CFS
      R013 = R0013
      R023 = R0023*CFS + R0033*SFS
      R033 =-R0023*SFS + R0033*CFS
C
C TRANSFORMATION FROM SAX TO BAX FINAL POSITION, FOR USE IN AERO
C COMPUTATIONS. NOT SUITABLE FOR INERTAL LOADS, BECAUSE A LINEAR
C MODEL FOR THE ELASTIC ROTATIONS HAS BEEN USED.
C
      R11 =      R011 + EA3*R021 - EA2*R031
      R21 = -EA3*R011 +     R021 + EA1*R031
      R31 =  EA2*R011 - EA1*R021 +     R031
      R12 =      R012 + EA3*R022 - EA2*R032
      R22 = -EA3*R012 +     R022 + EA1*R032
      R32 =  EA2*R012 - EA1*R022 +     R032
      R13 =      R013 + EA3*R023 - EA2*R033
      R23 = -EA3*R013 +     R023 + EA1*R033
      R33 =  EA2*R013 - EA1*R023 +     R033
C
C BRL POSITION ENTITIES.
C
      BP1 = X1 + EB1
      BP2 = X2 + EB2
      BP3 = X3 + EB3
```

```
C
      BD1 = XD1 + EDB1
      BD2 = XD2 + EDB2
      BD3 = XD3 + EDB3
C
      B1 = BP1 + B001
      B2 = BP2 + B002
      B3 = BP3 + B003
C
C SOLVE FOR IX.
C
      DUM = .5 * (EA2*EA2 + EA3*EA3)
C
      IX1 = DUM*R0011
      IX2 = DUM*R0012
      IX3 = DUM*R0013
C
C SOLVE FOR IXD
C
      TEMP5 = -EA3*EDA3 - EA2*EDA2
      TEMP6 =  2.*EA1*EDA2
      TEMP7 =  2.*EA1*EDA3
C
      IXD1 = R0011*TEMP5 + R0021*TEMP6 + R0031*TEMP7
      IXD2 = R0012*TEMP5 + R0022*TEMP6 + R0032*TEMP7
      IXD3 = R0013*TEMP5 + R0023*TEMP6 + R0033*TEMP7
C
C SOLVE FOR IXDD.
C
      TEMP8  =   EDA3*EDA3 + .EDA2*EDA2 + EA3*EDDA3 + EA2*EDDA2
      TEMP9  = - EDA1*EDA2
      TEMP10 = - EDA1*EDA3
C
      IXDD1 = R0011*TEMP8 + R0021*TEMP9 + R0031*TEMP10
      IXDD2 = R0012*TEMP8 + R0022*TEMP9 + R0032*TEMP10
      IXDD3 = R0013*TEMP8 + R0023*TEMP9 + R0033*TEMP10
C
C 1.6 SHANK TO BLADE TRANSFORMATIONS
C
C AERO VELOCITY
C
C   CORRECTIONS TO THE AERO VELOCITY DUE TO INTERFERENCE EFFECTS. GET THE
C   AUGMENTIVE AERO VELOCITY, VAA(I), COMPUTED IN THE AIR MASS MODEL.
C
      CALL GETVAA ( J, VAA(I) )
C
C
      TEMP11 = VAS1 + BD1 + WAS2*B3 - WAS3*B2
      TEMP12 = VAS2 + BD2 + WAS3*B1 - WAS1*B3
      TEMP13 = VAS3 + BD3 + WAS1*B2 - WAS2*B1
C
      VA1 = R11*TEMP11 + R12*TEMP12 + R13*TEMP13 + VAA1
      VA2 = R21*TEMP11 + R22*TEMP12 + R23*TEMP13 + VAA2
      VA3 = R31*TEMP11 + R32*TEMP12 + R33*TEMP13 + VAA3
C
```

```
C
C 1.7 DISTRIBUTED INERTIAL LOADS
C
      TEMP14 = 2. * (-WSMZ3*BD2 + WSMZ2*BD3 + Z*XD2 )
      TEMP15 = 2. * ( WSMZ3*BD1 - WSMZ1*BD3 - Z*XD1 )
      TEMP16 = 2. * ( WSMZ1*BD2 - WSMZ2*BD1 )
C
      UMBR1 = US1 + XDD1 + TEMP14 + WDS2*B3 - WDS3*B2
     @       + U11*B1 + U12*B2 + U13*B3 - ZSQ*X1
C
      UMBR2 = US2 + XDD2 + TEMP15 + WDS3*B1 - WDS1*B3
     @       + U12*B1 + U22*B2 + U23*B3 - ZSQ*X2
C
      UMBR3 = US3+XDD3 + TEMP16 + WDS1*B2 - WDS2*B1
     @       + U13*B1 + U23*B2 + U33*B3
C
      FMI1 = - MB*UMBR1
      FMI2 = - MB*UMBR2
      FMI3 = - MB*UMBR3
C
C
C 1.8  AERO GEOMETRY
C
      M2  = VA2*RVC
      M3  = VA3*RVC
C
C COMPUTE M, SAB, AND CAB BY USING M2 AND M3.
C
      CALL TRIGAR (M2,M3,M,SAB,CAB)
C
C
C 1.9  AERODYNAMIC COEFFICIENT TABLES
C
      CALL LOOKUP (SAB,CAB,CFA2,CFA3,CMA1)
C
C 1.10 DISTRIBUTED AIR LOADS
C
      QAC = 1481. * M * M * CB * PR
      FABC2 = - QAC * CFA2
      FABC3 = - QAC * CFA3
C
C TIP LOSS CORRECTION: EQUIVALENT TO AN EFFECTIVE RADIUS OF 97 %.
C
      IF (TIPLOS .EQ. 0) GO TO 1110
      IF (J .NE. 11) GO TO 1110
      FABC3 = .4 * FABC3
 1110 CONTINUE
C
      RR12 =  R21*CAB + R31*SAB
      RR13 = -R21*SAB + R31*CAB
      RR22 =  R22*CAB + R32*SAB
      RR23 = -R22*SAB + R32*CAB
      RR32 =  R23*CAB + R33*SAB
      RR33 = -R23*SAB + R33*CAB
C
```

```
      FMA1 = RR12*FABC2 + RR13*FABC3 - FA01
      FMA2 = RR22*FABC2 + RR23*FABC3 - FA02
      FMA3 = RR32*FABC2 + RR33*FABC3 - FA03
      FMA4 = QAC * CMA1 * CB
C
C 1.11 LOADS INTEGRAND ASSEMBLER.
C
        FMB1 = FMI1 + FMA1
        FMB2 = FMI2 + FMA2
        FMB3 = FMI3 + FMA3
C
        IGR1 = GP11*FMB1 + GP12*FMB2 + GP13*FMB3
        IGR2 = GP21*FMB1 + GP22*FMB2 + GP23*FMB3
        IGR3 = GP34*FMA4
C
        IFR1 = FP11*FMB1 + FP12*FMB2 + FP13*FMB3
        IFR2 = FP21*FMB1 + FP22*FMB2 + FP23*FMB3
        IFR3 = FP31*FMB1 + FP32*FMB2 + FP33*FMB3
C
        TEMP21 = EDDB1 + 2.*Z*EDB2 - ZSQ*EB1
        TEMP22 = EDDB2 - 2.*Z*EDB1 - ZSQ*EB2
C
        DUM8 = - MB*TEMP21
        DUM9 = - MB*TEMP22
        DUM10= - MB*EDDB3
C
        TEMP23 = MB*Z0*Z0*B001 + FA01
        TEMP24 = MB*Z0*Z0*B002 + FA02
C
        IMR1 = MP11*FMB1 + MP12*FMB2 + MP13*FMB3
     @       + BP2*FMB3 - BP3*FMB2   - BP3*DUM9 + BP2*DUM10
     @       + X2*FA03 - X3*TEMP24
C
        IMR2 = MP21*FMB1 + MP22*FMB2 + MP23*FMB3
     @       + BP3*FMB1 - BP1*FMB3   - BP1*DUM10 + BP3*DUM8
     @       + X3*TEMP23 - X1*FA03
C
        IMR3 = MP31*FMB1 + MP32*FMB2 + MP33*FMB3
     @       + BP1*FMB2 - BP2*FMB1 + BP1*DUM9 - BP2*DUM8
     @       + X1*TEMP24 - X2*TEMP23
C
C 1.12 RADIAL SUMMATIONS.
C
        X1 = X1 + DS * IX1
        X2 = X2 + DS * IX2
        X3 = X3 + DS * IX3
C
        XD1 = XD1 + DS * IXD1
        XD2 = XD2 + DS * IXD2
        XD3 = XD3 + DS * IXD3
C
        XDD1 = XDD1 + DS * IXDD1
        XDD2 = XDD2 + DS * IXDD2
        XDD3 = XDD3 + DS * IXDD3
C
```

```
         GRI1 = GRI1 + DS*IGR1
         GRI2 = GRI2 + DS*IGR2
         GRI3 = GRI3 + DS*IGR3
C
         FRI1 = FRI1 + DS*IFR1
         FRI2 = FRI2 + DS*IFR2
         FRI3 = FRI3 + DS*IFR3
C
         MRI1 = MRI1 + DS*IMR1
         MRI2 = MRI2 + DS*IMR2
         MRI3 = MRI3 + DS*IMR3
C
C        TEST FOR COMPLETION OF RADIAL STATIONS.
C
         IF (J .LT. NR) GO TO 9002
C
C*****************  END OF RADIAL LOOP  *********************
C
C-----   SAVE FORCES AND MOMENTS OF BLADE # 1 FOR
C        COMPUTATION OF ROOT LOADS IN SHANK AXES.
C
         IF (I .GT. 1) GO TO 1130
         FTEM1 = FRI1
         FTEM2 = FRI2
         FTEM3 = FRI3
         MTEM1 = MRI1
         MTEM2 = MRI2
         MTEM3 = MRI3
C
C
C 1.13 MODAL COORDINATE INTEGRATIONS
C
 1130 CONTINUE
C
         DUM11=ZP*QDI1
         DUM12=ZP*QDI2
         DUM13=ZP*QDI3
C
         DUM21=ZP*QI1
         DUM22=ZP*QI2
         DUM23=ZP*QI3
C
         TEMP25=ZZ*QI1
         TEMP26=ZZ*QI2
         TEMP27=ZZ*QI3
C
C GENERALIZED MODAL EXCITATIONS
C
         GI1=GRI1+GV11*QDI1+GZQ11*TEMP25+GZV11*DUM11
      @          +GV12*QDI2+GZQ12*TEMP26+GZV12*DUM12
      @          +GV13*QDI3+GZQ13*TEMP27+GZV13*DUM13+ZZ*GBO1
C
         GI2=GRI2+GV21*QDI1+GZQ21*TEMP25+GZV21*DUM11
      @          +GV22*QDI2+GZQ22*TEMP26+GZV22*DUM12
      @          +GV23*QDI3+GZQ23*TEMP27+GZV23*DUM13+ZZ*GBO2
```

```
C
      GI3=GRI3+GV31*QDI1+GZQ31*TEMP25+GZV31*DUM11
     @          +GV32*QDI2+GZQ32*TEMP26+GZV32*DUM12
     @          +GV33*QDI3+GZQ33*TEMP27+GZV33*DUM13+ZZ*GBO3
C
      DUM24=ZP*GI1
      DUM25=ZP*GI2
      DUM26=ZP*GI3
C
C MODE # 1 ENTITIES
C
      QSI1  = QI1
      QI1   = KQQ11*QSI1    + KQV11*QDI1    + KQG11*GI1
     @      + KQZQ11*DUM21 + KQZV11*DUM11 + KQZG11*DUM24
      QDI1  = KVQ11*QSI1    + KVV11*QDI1    + KVG11*GI1
     @      + KVZQ11*DUM21 + KVZV11*DUM11 + KVZG11*DUM24
      QDDI1 = GI1 - ESQQ11*QI1
C
C MODE # 2 ENTITIES
C
      QSI2  = QI2
      QI2   = KQQ22*QSI2    + KQV22*QDI2    + KQG22*GI2
     @      + KQZQ22*DUM22 + KQZV22*DUM12 + KQZG22*DUM25
      QDI2  = KVQ22*QSI2    + KVV22*QDI2    + KVG22*GI2
     @      + KVZQ22*DUM22 + KVZV22*DUM12 + KVZG22*DUM25
      QDDI2 = GI2 - ESQQ22*QI2
C
C MODE # 3 ENTITIES
C
      QSI3  = QI3
      QI3   = KQQ33*QSI3    + KQV33*QDI3    + KQG33*GI3
     @      + KQZQ33*DUM23 + KQZV33*DUM13 + KQZG33*DUM26
      QDI3  = KVQ33*QSI3    + KVV33*QDI3    + KVG33*GI3
     @      + KVZQ33*DUM23 + KVZV33*DUM13 + KVZG33*DUM26
      QDDI3 = 0
C
C SAVE MODAL STATE VARIABLES
C
      CALL STORQS(I)
C
C
C 1.14 MULTIBLADE MODAL COMBINATIONS
C
      SYBION=RNB*SYBI
      CYBION=RNB*CYBI
C
      QS1=QS1+QI1*SYBION
      QC1=QC1+QI1*CYBION
      Q01=Q01+QI1*RNB
C
      QDS1=QDS1+QDI1*SYBION
      QDC1=QDC1+QDI1*CYBION
      QD01=QD01+QDI1*RNB
C
      QS2=QS2+QI2*SYBION
      QC2=QC2+QI2*CYBION
      Q02=Q02+QI2*RNB
C
      QDS2=QDS2+QDI2*SYBION
      QDC2=QDC2+QDI2*CYBION
      QD02=QD02+QDI2*RNB
C
      QS3=QS3+QI3*SYBION
      QC3=QC3+QI3*CYBION
      Q03=Q03+QI3*RNB
C
      QDS3=QDS3+QDI3*SYBION
      QDC3=QDC3+QDI3*CYBION
      QD03=QD03+QDI3*RNB
C
C 1.15 MULTIBLADE LOADS COMBINATIONS
C
      FGR1 = FGR1 + CYBION*FRI1 + SYBION*FRI2
      FGR2 = FGR2 - SYBION*FRI1 + CYBION*FRI2
      FGR3 = FGR3 + RNB*FRI3
C
      MGR1 = MGR1 + CYBION*MRI1 + SYBION*MRI2
      MGR2 = MGR2 - SYBION*MRI1 + CYBION*MRI2
      MGR3 = MGR3 + RNB*MRI3
C
C
C     TEST FOR COMPLETION OF BLADES.
C
*     IF (I .LT. NB) GO TO 9001
C
C*****************  END OF BLADE LOOP  ***********************
C
C
C 1.16 GIMBAL LOADS SUMMATION
C
      FSQ11=FQ11+FZQ11*ZZ
      FSQ21=FQ21+FZQ21*ZZ
      FSQ31=FQ31+FZQ31*ZZ
C
      FSQD11=FQD11+FZQD11*ZP
      FSQD21=FQD21+FZQD21*ZP
      FSQD31=FQD31+FZQD31*ZP
C
      MSQ11=MQ11+MZQ11*ZZ
      MSQ21=MQ21+MZQ21*ZZ
      MSQ31=MQ31+MZQ31*ZZ
C
      MSQD11=MQD11+MZQD11*ZP
      MSQD21=MQD21+MZQD21*ZP
      MSQD31=MQD31+MZQD31*ZP
C
      FSQ12=FQ12+FZQ12*ZZ
      FSQ22=FQ22+FZQ22*ZZ
      FSQ32=FQ32+FZQ32*ZZ
C
```

```
      FSQD12=FQD12+FZQD12*ZP
      FSQD22=FQD22+FZQD22*ZP
      FSQD32=FQD32+FZQD32*ZP
C
      MSQ12=MQ12+MZQ12*ZZ
      MSQ22=MQ22+MZQ22*ZZ
      MSQ32=MQ32+MZQ32*ZZ
C
      MSQD12=MQD12+MZQD12*ZP
      MSQD22=MQD22+MZQD22*ZP
      MSQD32=MQD32+MZQD32*ZP
C
      FSQ13=FQ13+FZQ13*ZZ
      FSQ23=FQ23+FZQ23*ZZ
      FSQ33=FQ33+FZQ33*ZZ
C
      FSQD13=FQD13+FZQD13*ZP
      FSQD23=FQD23+FZQD23*ZP
      FSQD33=FQD33+FZQD33*ZP
C
      MSQ13=MQ13+MZQ13*ZZ
      MSQ23=MQ23+MZQ23*ZZ
      MSQ33=MQ33+MZQ33*ZZ
C
      MSQD13=MQD13+MZQD13*ZP
      MSQD23=MQD23+MZQD23*ZP
      MSQD33=MQD33+MZQD33*ZP
C
      FS01 = F01 + FZ1*ZZ
      FS02 = F02 + FZ2*ZZ
      FS03 = F03 + FZ3*ZZ
      MS01 = M01 + MZ1*ZZ
      MS02 = M02 + MZ2*ZZ
      MS03 = M03 + MZ3*ZZ
C
      FG1=FGR1+FSQ11*QC1+FSQ21*QS1 +FSQD11*QDC1+FSQD21*QDS1
     @       + FSQ12*QC2+FSQ22*QS2 +FSQD12*QDC2+FSQD22*QDS2
     @        +FSQ13*QC3+FSQ23*QS3 +FSQD13*QDC3+FSQD23*QDS3
C
      FG2=FGR2+FSQ21*QC1 -FSQ11*QS1+FSQD21*QDC1-FSQD11*QDS1
     @        +FSQ22*QC2 -FSQ12*QS2+FSQD22*QDC2-FSQD12*QDS2
     @        +FSQ23*QC3 -FSQ13*QS3+FSQD23*QDC3+FSQD13*QDS3
C
      FG3=FGR3+FSQ31*Q01+FSQD31*QD01
     @        +FSQ32*Q02+FSQD32*QD02
     @        +FSQ33*Q03+FSQD33*QD03 + FS03
C
      MG1=MGR1+MSQ11*QC1+MSQ21*QS1+MSQD11*QDC1+MSQD21*QDS1
     @        +MSQ12*QC2+MSQ22*QS2+MSQD12*QDC2+MSQD22*QDS2
     @        +MSQ13*QC3+MSQ23*QS3+MSQD13*QDC3+MSQD23*QDS3
C
      MG2=MGR2+MSQ21*QC1-MSQ11*QS1+MSQD21*QDC1-MSQD11*QDS1
     @        +MSQ22*QC2-MSQ12*QS2+MSQD22*QDC2-MSQD12*QDS2
     @        +MSQ23*QC3-MSQ13*QS3+MSQD23*QDC3-MSQD13*QDS3
C
```

```
         MG3=MGR3+MSQ31*Q01+MSQD31*QD01
     @           +MSQ32*Q02+MSQD32*QD02
     @           +MSQ33*Q03+MSQD33*QD03
C
C--------------------------------------------
C COMPUTE ROOT LOADS OF BLADE # 1 IN SHANK AXES
C--------------------------------------------
C
         FBLD1 = FTEM1 + FSQ11*QI1A + FSQD11*QDI1A
     @                 + FSQ12*QI2A + FSQD12*QDI2A
     @                 + FSQ13*QI3A + FSQD13*QDI3A
C
         FBLD2 = FTEM2 + FSQ21*QI1A + FSQD21*QDI1A
     @                 + FSQ22*QI2A + FSQD22*QDI2A
     @                 + FSQ23*QI3A + FSQD23*QDI3A + FS02
C
         FBLD3 = FTEM3 + FSQ31*QI1A + FSQD31*QDI1A
     @                 + FSQ32*QI2A + FSQD32*QDI2A
     @                 + FSQ33*QI3A + FSQD33*QDI3A + FS03
C
         MBLD1 = MTEM1 + MSQ11*QI1A + MSQD11*QDI1A
     @                 + MSQ12*QI2A + MSQD12*QDI2A
     @                 + MSQ13*QI3A + MSQD13*QDI3A + MS01
C
         MBLD2 = MTEM2 + MSQ21*QI1A + MSQD21*QDI1A
     @                 + MSQ22*QI2A + MSQD22*QDI2A
     @                 + MSQ23*QI3A + MSQD23*QDI3A + MS02
C
         MBLD3 = MTEM3 + MSQ31*QI1A + MSQD31*QDI1A
     @                 + MSQ32*QI2A + MSQD32*QDI2A
     @                 + MSQ33*QI3A + MSQD33*QDI3A + MS03
C
C
C 1.17 GIMBAL TO HUB TRANSFORMATIONS
C
         FCU1 = CYC*FG1 + SYC*FG2
         FCU2 =-SYC*FG1 + CYC*FG2
C
         MCU1 = CYC*MG1+SYC*MG2 - ZU*FCU2
         MCU2 =-SYC*MG1+CYC*MG2 + ZU*FCU1
C
         EG1 = MCU2
C
         FT1 = FCU1 + QG1*FG3
C
         MT1 = MCU1 + QG1*MG3 + ZU*FCU2
         MT2 = MCU2 - ZU*FT1
C
         FH1 = CYT*FT1 - SYT*FCU2
         FH2 = SYT*FT1 + CYT*FCU2
         FH3 = FG3 - QG1*FCU1
C
         MH1 = CYT*MT1 - SYT*MT2
         MH2 = SYT*MT1 + CYT*MT2
         MH3 = MG3 - QG1*MCU1
```

```
C
C 1.18 GIMBAL SERVO INTEGRATOR.
C
        QDDG1=KGE*EG1+KGQC*QC1+KGQS*QS1
     @            +KGVC*QDC1+KGVS*QDS1
C
        QSG1=QG1
C
        DUM14=ZP*QDG1
        DUM15=ZP*QDDG1
        DUM16=ZP*QSG1
        DUM17=ZP*QDDG1
C
        QG1=GQQ*QSG1+GQV*QDG1+GQZV*DUM14
     @            +GQA*QDDG1+GQZA*DUM15
C
        QDG1=GVV*QDG1+GVQ*QSG1+GVZQ*DUM16
     @            +GVA*QDDG1+GVZA*DUM17
C
C
C     END THE ROTOR MODEL.
C
```

C-3

```
C
C 2.0 SUPPORT SYSTEM EQUATIONS.
C
        IF (TOWER .EQ. 0) GO TO 2010
C
C MODEL WITH THREE NORMAL MODES FOR SURGE, SWAY, AND YAW(TORSION).
C
C GENERALIZED FORCING FUNCTIONS
C
        GS1 = GSF11*FH1 + GSF13*FH3 + GSM12*MH2
        GS2 = GSF22*FH2 + GSM23*MPOD
        GS3 = GSF32*FH2 + GSM31*MH1
C
C MODAL ACCELERATIONS
C
        SDD1 = GS1 - DS11*SD1 - W2S11*S1
        SDD2 = GS2 - DS22*SD2 - W2S22*S2
        SDD3 = GS3 - DS33*SD3 - W2S33*S3
C
C NUMERICAL INTEGRATION OF MODAL COORDINATES
C
        SD1 = SD1 + SDD1*DT
        SD2 = SD2 + SDD2*DT
        SD3 = SD3 + SDD3*DT
C
        S1  = S1 + SD1*DT
        S2  = S2 + SD2*DT
        S3  = S3 + SD3*DT
C
C TRANSLATIONAL VELOCITY
C
        VH1 = VBH1 + SV11*SD1
        VH2 = VBH2 + SV22*SD2 + SV23*SD3
        VH3 = VBH3 + SV31*SD1
C
C ROTATIONAL VELOCITY
C
        WH1 = WBH1 + SW13*SD3
        WH2 = WBH2 + SW21*SD1
        WH3 = WBH3 + SW32*SD2
C
C TIME DERIVATIVE OF TRANSLATIONAL VELOCITY
C
        VDH1 = VDBH1 + SV11*SDD1
        VDH2 = VDBH2 + SV22*SDD2 + SV23*SDD3
        VDH3 = VDBH3 + SV31*SDD1
C
C TIME DERIVATIVE OF ROTATIONAL VELOCITY
C
        WDH1 = WDBH1 + SW13*SDD3
        WDH2 = WDBH2 + SW21*SDD1
        WDH3 = WDBH3 + SW32*SDD2
C
 2010   CONTINUE
C
```

```
C
C 3.0-CONTROL SYSTEM
C
        IF (CONTRL .EQ. 0) GO TO 3010
C
        ZERR = SRQ - Z
        AD1  = AZ1 * ZERR
        AD2 = AA21*A1  - AA22*A2  + AZ2 * ZERR
        AD3  = AG3*ZA + AD03
C
C NUMERICAL INTEGRATION
C
        A1  = A1 + DT * AD1
        A2  = A2 + DT * AD2
        A3  = A3 + DT * AD3
C
C RESULTS
C
        C1  = CA12*A2
        C2  = 0
        C3  = 0
C
 3010   CONTINUE
C
        LGC = LA13*A3 + LPR*PRQ + LG*ZA
C
C
C 4.0 - POWER TRAIN
C
        IF (PWRTRN .EQ. 0) GO TO 4010
C
C ACCELERATIONS.
C
        PDD1 = -KP12*P2 -BP11*PD1 -BP12*PD2 +PM1*MH3 +PZD1*ZD
        PDD2 = -KP22*P2 +BP21*PD1 -BP22*PD2 +PM2*MH3 +PZD2*ZD +PL2*LGC
C
C NUMERICAL INTEGRATION.
C
        PD1 = PD1 + PDD1 * DT
        PD2 = PD2 + PDD2 * DT
        P2  = P2  + PD2  * DT
C
C OUTPUTS
C
        ZD  = PDD1
        Z   = PD1
        ZP  = Z/Z0 - 1.
        ZA  = PD1 - PD2
C
C COMPUTE THE MOMENT APPLIED BY THE POWER TRAIN ON THE TOWER
C
C       MPOD = .... DEPENDS ON THE SPECIFICS OF THE POWER TRAIN ....
C
 4010   CONTINUE
C
```

```
C RATE RESOLVER:
C
C
      SDY = Z * DT
C
C COMPUTE AMPLITUDE ERROR
C
      EA = SY*SY + CY*CY - 1.
C
C MAKE CORRECTION
C
      CDY = CDY - .25*EA
C
C SET UP ADVANCE
C
      CYSAV = CY
      CY    = CY * CDY - SY * SDY
      SY    = SY * CDY + CYSAV*SDY
C
C
C 5.0 AIR MASS.
C
C RETARDATION VELOCITY
C
        IF (RETARD .EQ. 0) GO TO 5010
C
C   ERROR FUNCTION
C
        EVR = 1. - RV * RV * (VAH2*VAH2 + VAH3*VAH3)
C
C   CORRECTION SERVO
C
        RV = RV + EVR*GCEV
C
C   RETARDATION
C
        VRW = KAR * FH3 * RV * RHOR
C
 5010   CONTINUE
C
C
C WINDGUST RANDOM SIGNALS
C
        IF (GUSTS .EQ. 0) GO TO 5020
C
C   EQUATIONS WHICH FOLLOW USE A RANDOM NUMBER GENERATOR WHICH
C   APPEARS AS A WHITE NOISE SOURCE TO EACH FILTER EQUATION.
C
        QDDGU1=RANDOM(IGNORE) -ZTG11*WGG11*QDGU1 -WGG11*WGG11*QGU1
        QDDGU2=RANDOM(IGNORE) -ZTG22*WGG22*QDGU2 -WGG22*WGG22*QGU2
        QDDGU3=RANDOM(IGNORE) -ZTG33*WGG33*QDGU3 -WGG33*WGG33*QGU3
        QDDGU4=RANDOM(IGNORE) -ZTG44*WGG44*QDGU4 -WGG44*WGG44*QGU4
        QDDGU5=RANDOM(IGNORE) -ZTG55*WGG55*QDGU5 -WGG55*WGG55*QGU5
        QDDGU6=RANDOM(IGNORE) -ZTG66*WGG66*QDGU6 -WGG66*WGG66*QGU6
C
```

```
C   NUMERICAL INTEGRATION OF WINDGUST FILTERS.
C
       QDGU1=QDGU1+QDDGU1*DT
       QDGU2=QDGU2+QDDGU2*DT
       QDGU3=QDGU3+QDDGU3*DT
       QDGU4=QDGU4+QDDGU4*DT
       QDGU5=QDGU5+QDDGU5*DT
       QDGU6=QDGU6+QDDGU6*DT
C
       QGU1=QGU1+QDGU1*DT
       QGU2=QGU2+QDGU2*DT
       QGU3=QGU3+QDGU3*DT
       QGU4=QGU4+QDGU4*DT
       QGU5=QGU5+QDGU5*DT
       QGU6=QGU6+QDGU6*DT
C
C   OUTPUTS
C
       VG1=VGU11*QDGU1
       VG2=VGU22*QDGU2
       VG3=VGU33*QDGU3
C
       WG1=WGU11*QDGU4
       WG2=WGU22*QDGU5
       WG3=WGU33*QDGU6
C
 5020  CONTINUE
C
C PRIMARY WIND SIGNALS
C
       VWT1=VW1+VG1
       VWT2=VW2+VG2
       VWT3=VW3+VG3
C
       WWT1=WW1+WG1
       WWT2=WW2+WG2
       WWT3=WW3+WG3
C
       VWT1=VWT1+VRW
C
C
C INTERFERENCE EFFECTS:
C FOR EACH BLADE, COMPUTE THE AUGMENTIVE AERO VELOCITY DUE TO INTERFERENCE
C EFFECTS AS A FUNTION OF THE RADIAL STATION. IN THE CASE OF TOWER SHADOW
C AND WINDSHEAR EFFECTS, THIS COMPUTATION CAN BE MORE CONVENIENTLY DONE
C WITHIN THE RADIAL LOOP.
C
       CALL STOVAA ( SY, CY )
C
C PASS THROUGH MODELS IS COMPLETE
C
C************  BOTTOM OF INFINITE LOOP  ********************
C
       GO TO 9000
```

APPENDIX H

SYSTEM EQUATIONS OF THE WIND TURBINE SIMULATION MODEL

Presented in this Appendix are the system equations which constitute the WEST-3 wind turbine simulation model, i.e., these equations are the normalized and scaled versions of those given in Appendix G.

```
C
C***************************************************************
C
C WEST-3 SYSTEM EQUATIONS: JUNE 1985
C
C***************************************************************
C
C
C INITIALIZATIONS
C
      NB = 2
      NR = 11
C
      Z = ZINT
      ZD= 0
C
      CY = 1.
      SY = 0
      CDY= .5
C
      VH1= VBH1
      VH2= VBH2
      VH3= VBH3
      WH1= WBH1
      WH2= WBH2
      WH3= WBH3
C
      VDH1= VDBH1
      VDH2= VDBH2
      VDH3= VDBH3
      WDH1= WDBH1
      WDH2= WDBH2
      WDH3= WDBH3
C
      VWT1 = VW1
      VWT2 = VW2
      VWT3 = VW3
      WWT1 = WW1
      WWT2 = WW2
      WWT3 = WW3
      VRW  = 0
C
      S1 = 0
      S2 = 0
      S3 = 0
      SD1= 0
      SD2= 0
      SD3= 0
      MPOD=0
C
      C1 = 0
      A1 = 0
      A2 = 0
      A3 = 0
      ZA = 0
```

```
C
        PD1= Z
        P2 = 0
        PD2= 0
        MPOW = 0
C
        QI1A  = 0
        QDI1A = 0
        QDDI1A= 0
        QI2A  = 0
        QDI2A = 0
        QDDI2A= 0
        QI1B  = 0
        QDI1B = 0
        QDDI1B= 0
        QI2B  = 0
        QDI2B = 0
        QDDI2B= 0
C
C
C***************** BEGINING OF INFINITE LOOP  *************************
C
 9000 CONTINUE
C
C
C AEROELASTIC ROTOR MODEL
C
C ROTOR SPEED VARIABLES
C
        ZP= MULT2 (Z-.5)
        ZZ = .5*ZP*ZP + ZP
        ZSQ = Z * Z
C
C 1.2 HUB TO SHANK TRANSFORMATIONS
C
C     MANY OF THE TRANSFORMATIONS ARE COMMON TO ALL THE
C     BLADES, AND ARE PROCESSED OUTSIDE THE BLADE LOOP.
C
C GEOMETRY
C
        SYT = SYC*CY - CYC*SY
        CYT = CYC*CY + SYC*SY
C
C INERTIAL VELOCITY
C
        WTUMZ1 = CYT*WH1 + SYT*WH2
        WTUMZ2 =-SYT*WH1 + CYT*WH2
        WTUMZ3 = WH3
        VTU1 = CYT*VH1 + SYT*VH2 + .4*ZU*WTUMZ2
        VTU2 =-SYT*VH1 + CYT*VH2 - .4*ZU*WTUMZ1
        VTU3 = VH3
C
        WCMZ1 = WTUMZ1 - .1*WTUMZ3*QG1 + MULT2(Z*QG1)
        WCMZ2 = WTUMZ2 + QDG1
        WCMZ3 = WTUMZ3 + .4*WTUMZ1*QG1
```

```
      VC1 = VTU1 - .2*VTU3*QG1 - .4*ZU*WCMZ2
      VC2 = VTU2 + .4*ZU*WCMZ1
      VC3 = VTU3 + .2*VTU1*QG1
C
C AERO VELOCITY
C
      WAH1 = .5 *WH1 + WWT1*CYR2 - WWT3*SYR2
      WAH2 = .5 *WH2 + WWT2
      WAH3 = .25*WH3 + WWT1*SYR2 + WWT3*CYR2
      VAH1 = .125*VH1 + VWT1*CYR2 - VWT3*SYR2
      VAH2 = .125*VH2 + VWT2
      VAH3 = .125*VH3 + VWT1*SYR2 + VWT3*CYR2
C
      WATU1 = CYT*WAH1 + SYT*WAH2
      WATU2 =-SYT*WAH1 + CYT*WAH2
      WATU3 = .2 *WAH3 - Z
      VATU1 = CYT*VAH1 + SYT*VAH2 + .1*ZU*WATU2
      VATU2 =-SYT*VAH1 + CYT*VAH2 - .1*ZU*WATU1
      VATU3 = VAH3
C
      WAC1 = WATU1 - WATU3*QG1
      WAC2 = WATU2 + .5*QDG1
      WAC3 = WATU3 + .04*WATU1*QG1
      VAC1 = VATU1 - .2*VATU3*QG1 - .1*ZU*WAC2
      VAC2 = VATU2 + .1*ZU*WAC1
      VAC3 = VATU3 + .2*VATU1*QG1
C
C TIME DERIVATIVE OF INERTIAL VELOCITY.
C
      DUM1  = VDH1 - Z*VH2
      DUM2  = VDH2 + Z*VH1
      TEMP1 = WDH1 - Z*WH2
      TEMP2 = WDH2 + Z*WH1
C
      WDTU1 = CYT*TEMP1 + SYT*TEMP2
      WDTU2 =-SYT*TEMP1 + CYT*TEMP2
      WDTU3 = WDH3-ZD
      VDTU1 = CYT*DUM1 + SYT*DUM2 + .4*ZU*WDTU2
      VDTU2 =-SYT*DUM1 + CYT*DUM2 - .4*ZU*WDTU1
C
      DUM3 = WDTU1 - .05*WTUMZ3*QDG1 + Z*QDG1
      DUM4 = WDTU3 + .2 *WTUMZ1*QDG1
      TEMP3= VDTU1 - .1*VTU3*QDG1
      TEMP4= VDH3  + .1*VTU1*QDG1
C
      WDC1 = DUM3  - .1*DUM4*QG1
      WDC2 = WDTU2 + .5*QDDG1
      WDC3 = DUM4  + .4*DUM3*QG1
      VDC1 = TEMP3 - .2*TEMP4*QG1 - .4*ZU*WDC2
      VDC2 = VDTU2 + .4*ZU*WDC1
      VDC3 = TEMP4 + .2*TEMP3*QG1
C
C  INITIALIZATIONS FOR BLADE LOOP
C
      I = 0
```

```
        LOOPI = 0
C
        SYBI =-SYB
        CYBI = CYB
C
C     MULTIBLADE COORDINATE AND LOADS.
C
        QS1=0
        QC1=0
        Q01=0
        QDS1=0
        QDC1=0
        QD01=0
C
        QS2=0
        QC2=0
        Q02=0
        QDS2=0
        QDC2=0
        QD02=0
C
        QS3=0
        QC3=0
        Q03=0
        QDS3=0
        QDC3=0
        QD03=0
C
        FGR1=0
        FGR2=0
        FGR3=0
C
        MGR1=0
        MGR2=0
        MGR3=0
C
C*****************   BEGINING OF BLADE LOOP  ***********************
C
C 1.1  BLADE NUMBER SELECT
C
 9001   CONTINUE
C
        I = I + 1
        LOOPI = LOOPI + .25
C
        SYBISV = SYBI
        SYBI = SYB*CYBI + CYB*SYBI
        CYBI = CYB*CYBI - SYB*SYBISV
C
C LOAD QI,QDI AND QDDI FOR BLADE I FROM
C SPACE CONTAINING Q, QD AND QDD FOR ALL BLADES.
C
C       CALL LOADQS(I)
C
C THE FOLLOWING SEQUENCE IMPLEMENTS THE "LOADQS" FUNCTION FOR 2 BLADES
```

```
C
  201   IF (I .GT. 1) GO TO 202
        QI1=QI1A
        QDI1=QDI1A
        QDDI1=QDDI1A
        QI2=QI2A
        QDI2=QDI2A
        QDDI2=QDDI2A
        QI3=QI3A
        QDI3=QDI3A
        QDDI3=QDDI3A
        GO TO 203
C
  202   CONTINUE
        QI1=QI1B
        QDI1=QDI1B
        QDDI1=QDDI1B
        QI2=QI2B
        QDI2=QDI2B
        QDDI2=QDDI2B
        QI3=QI3B
        QDI3=QDI3B
        QDDI3=QDDI3B
C
  203 CONTINUE
C
C
C 1.2 HUB TO SHANK TRANSFORMATIONS
C
C     THESE TRANSFORMATIONS DEPEND ON THE BLADE AZIMUTHAL POSITION
C
C GEOMETRY
C
        CYI = CY*CYBI - SY*SYBI
        SYI = SY*CYBI + CY*SYBI
C
        SYCBI = SYC*CYBI + CYC*SYBI
        CYCBI = CYC*CYBI - SYC*SYBI
C
C BLADE FEATHERING CONTROL ANGLE
C
        F = C1 - .1*C2*CYI - .1*C3*SYI
C
C AERO VELOCITY
C
        WAS1 = CYCBI*WAC1 - SYCBI*WAC2
        WAS2 = SYCBI*WAC1 + CYCBI*WAC2
        WAS3 = WAC3
        VAS1 = CYCBI*VAC1 - SYCBI*VAC2
        VAS2 = SYCBI*VAC1 + CYCBI*VAC2
        VAS3 = VAC3
C
C INERTIAL VELOCITY
C
        WSMZ1 = CYCBI*WCMZ1 - SYCBI*WCMZ2
```

```
      WSMZ2 = SYCBI*WCMZ1 + CYCBI*WCMZ2
      WSMZ3 = WCMZ3
      VS1 = CYCBI*VC1 - SYCBI*VC2
      VS2 = SYCBI*VC1 + CYCBI*VC2
      VS3 = VC3
C
C TIME DERIVATIVE OF INERTIAL VELOCITY.
C
      WDS1 = CYCBI*WDC1 - SYCBI*WDC2
      WDS2 = SYCBI*WDC1 + CYCBI*WDC2
      WDS3 = WDC3
      VDS1 = CYCBI*VDC1 - SYCBI*VDC2
      VDS2 = SYCBI*VDC1 + CYCBI*VDC2
      VDS3 = VDC3
C
C ACCELERATION TERMS FOR USE IN SECTION 1.7
C
C MANY OF THE TERMS NEEDED FOR COMPUTING THE DISTRIBUTED
C INERTIA LOADS AT EACH BLADE RADIAL STATION ARE COMMON TO
C ALL THE STATIONS, AND ARE GENERATED OUTSIDE THE BLADE LOOP
C
      US1 = VDS1 +.1 *WSMZ2*VS3 -.05*WSMZ3*VS2 + Z*VS2
      US2 = VDS2 +.05*WSMZ3*VS1 -.1 *WSMZ1*VS3 - Z*VS1
      US3 = VDS3 +.1* WSMZ1*VS2 -.1 *WSMZ2*VS1
C
C THE PHYSICAL EQUATION FOR THE NEXT STATEMENT IS ... TERM = WSMZ3 - 2 * Z
C
      TERM   = .025*WSMZ3 - Z
      U11 = -WSMZ3*TERM  - .1*WSMZ2*WSMZ2
      U22 = -WSMZ3*TERM  - .1*WSMZ1*WSMZ1
      U33 = -WSMZ2*WSMZ2 - WSMZ1*WSMZ1
C
      U12 =     WSMZ2*WSMZ1
      U13 = .05*WSMZ3*WSMZ1 - Z*WSMZ1
      U23 = .05*WSMZ3*WSMZ2 - Z*WSMZ2
C
C     INITIALIZE FOR RADIAL INTEGRATION OF BLADE,I.
C
      J = 0
C
      X1=0
      X2=0
      X3=0
      XD1=0
      XD2=0
      XD3=0
      XDD1=0
      XDD2=0
      XDD3=0
C
      GRI1=0
      GRI2=0
      GRI3=0
C
      FRI1=0
```

```
       FRI2=0
       FRI3=0
C
       MRI1=0
       MRI2=0
       MRI3=0
C
C*****************   BEGINING OF RADIAL LOOP  ************************
C
C 1.3 RADIAL STATION COUNTER.
C
 9002 CONTINUE
      J = J + 1
C
C 1.4 RADIAL PHYSICAL PARAMETER BLOCKMOVE.  MOVE ALL BLADE PHYSICAL
C     PROPERTIES FROM A COMMON MEMORY REGION TO A SEPARATE REGION
C     APPLICABLE FOR A SINGLE BLADE ELEMENT.
C
C     ENTITIES MOVED (AND THEIR SIZES) ARE-
C
C     ROO(9),RFS,DA(3*NM),DB(3*NM),BOO(3),GP(4*NM),FP(9),MP(9),
C     FAO(3),MB,CB
C
C       CALL MOVIT(J)
C
C**************************************************************************
C
C THE FOLLOWING SEQUENCE IMPLEMENTS THE "MOVIT" FUNCTION.
C
C THIS SEQUENCE IS SET UP FOR AN 11 RADIAL STATION SITUATION.
C
  301   IF(J.GT.1)GO TO 302
        ROO11=ROO11A
        ROO21=ROO21A
        ROO31=ROO31A
        ROO12=ROO12A
        ROO22=ROO22A
        ROO32=ROO32A
        ROO13=ROO13A
        ROO23=ROO23A
        ROO33=ROO33A
C
        DA11=DA11A
        DA21=DA21A
        DA31=DA31A
        DA12=DA12A
        DA22=DA22A
        DA32=DA32A
        DA13=DA13A
        DA23=DA23A
        DA33=DA33A
C
        DB11=DB11A
        DB21=DB21A
        DB31=DB31A
      DB12=DB12A
      DB22=DB22A
      DB32=DB32A
      DB13=DB13A
      DB23=DB23A
      DB33=DB33A
C
      GP11=GP11A
      GP12=GP12A
      GP13=GP13A
      GP21=GP21A
      GP22=GP22A
      GP23=GP23A
      GP34=GP34A
C
      FP11=FP11A
      FP21=FP21A
      FP31=FP31A
      FP12=FP12A
      FP22=FP22A
      FP32=FP32A
      FP13=FP13A
      FP23=FP23A
      FP33=FP33A
C
      MP11=MP11A
      MP21=MP21A
      MP31=MP31A
      MP12=MP12A
      MP22=MP22A
      MP32=MP32A
      MP13=MP13A
      MP23=MP23A
      MP33=MP33A
C
      B001=B001A
      B002=B002A
      B003=B003A
C
      FA01=FA01A
      FA02=FA02A
      FA03=FA03A
C
      RFS=RFSA
      MB=MBA
      CB=CBA
C
      S  = SA
      DS = DSA
      GO TO 312
C
  302 IF(J.GT.2)GO TO 303
      R0011=R0011B
        .     .
        .     .
```

```
                .     .
              DS = DSB
              GO TO 312
C
   303        IF(J.GT.3)GO TO 304
              R0011=R0011C
                .     .
                .     .
                .     .
              DS = DSC
              GO TO 312
C
   304        IF(J.GT.4)GO TO 305
              R0011=R0011D
                .     .
                .     .
                .     .
              DS = DSD
              GO TO 312
C
   305        IF(J.GT.5)GO TO 306
              R0011=R0011E
                .     .
                .     .
                .     .
              DS = DSE
              GO TO 312
C
   306        IF(J.GT.6)GO TO 307
              R0011=R0011F
                .     .
                .     .
                .     .
              DS = DSF
              GO TO 312
C
   307        IF(J.GT.7)GO TO 308
              R0011=R0011G
                .     .
                .     .
                .     .
              DS = DSG
              GO TO 312
C
   308        IF(J.GT.8)GO TO 309
              R0011=R0011H
                .     .
                .     .
                .     .
              DS = DSH
              GO TO 312
C
   309        IF(J.GT.9)GO TO 310
              R0011=R0011I
                .     .
```

```
        •     •
        •     •
      DS = DSI
      GO TO 312
C
 310  IF(J.GT.10)GO TO 311
      R0011=R0011J
        •     •
        •     •
        •     •
      DS = DSJ
      GO TO 312
C
 311  IF(J.GT.11)GO TO 312
      R0011=R0011K
        •     •
        •     •
        •     •
      DS = DSK
C
 312 CONTINUE
C
C 1.5 BLADE ELEMENT GEOMETRY
C
C MODAL ANGULAR ACCELERATION: BAX WITH RESPECT TO SAX.
C
      EA1   = DA11*QI1   + DA12*QI2   + DA13*QI3
      EDA1  = DA11*QDI1  + DA12*QDI2  + DA13*QDI3
      EDDA1 = DA11*QDDI1 + DA12*QDDI2 + DA13*QDDI3
C
      EA2   = DA21*QI1   + DA22*QI2   + DA23*QI3
      EDA2  = DA21*QDI1  + DA22*QDI2  + DA23*QDI3
      EDDA2 = DA21*QDDI1 + DA22*QDDI2 + DA23*QDDI3
C
      EA3   = DA31*QI1   + DA32*QI2   + DA33*QI3
      EDA3  = DA31*QDI1  + DA32*QDI2  + DA33*QDI3
      EDDA3 = DA31*QDDI1 + DA32*QDDI2 + DA33*QDDI3
C
C LINEAR MODAL TRANSLATIONAL BAX MOVEMENT WITH RESPECT TO SAX.
C
      EB1   = DB11*QI1   + DB12*QI2   + DB13*QI3
      EDB1  = DB11*QDI1  + DB12*QDI2  + DB13*QDI3
      EDDB1 = DB11*QDDI1 + DB12*QDDI2 + DB13*QDDI3
C
      EB2   = DB21*QI1   + DB22*QI2   + DB23*QI3
      EDB2  = DB21*QDI1  + DB22*QDI2  + DB23*QDI3
      EDDB2 = DB21*QDDI1 + DB22*QDDI2 + DB23*QDDI3
C
      EB3   = DB31*QI1   + DB32*QI2   + DB33*QI3
      EDB3  = DB31*QDI1  + DB32*QDI2  + DB33*QDI3
      EDDB3 = DB31*QDDI1 + DB32*QDDI2 + DB33*QDDI3
C
C APPROXIMATIONS FOR SINE AND COSINE OF FEATHERING ANGLE, FS.
C IF "RFS" IS CONSTANT, THESE CAN BE OUTSIDE THE RADIAL LOOP.
C
```

```
        FS = RFS*F-RFS*FO
        FSQ = FS*FS
        SFS = MULT2 ( FS*(1.-.672*FSQ*(1.-.2*FSQ)) )
        CFS = 1. - MULT2(FSQ*(1.-.3333*FSQ))
C
C TRANSFORMATION FROM SAX TO BAX CONTROL POSITION
C
        R011 =    R0011
        R021 =    R0021*CFS +   R0031*SFS
        R031 =   -R0021*SFS +   R0031*CFS
        R012 =    R0012
        R022 = .5*R0022*CFS +.5*R0032*SFS
        R032 =   -R0022*SFS +   R0032*CFS
        R013 =    R0013
        R023 =    R0023*CFS +   R0033*SFS
        R033 =-.5*R0023*SFS +.5*R0033*CFS
C
C TRANSFORMATION FROM SAX TO BAX FINAL POSITION, FOR USE IN AERO
C COMPUTATIONS. NOT SUITABLE FOR INERTAL LOADS, BECAUSE A LINEAR
C MODEL FOR THE ELASTIC ROTATIONS HAS BEEN USED.
C
        R11 =          R011 +  .25*EA3*R021 -  .25*EA2*R031
        R21 =     -EA3*R011 +          R021 +  .5 *EA1*R031
        R31 =      EA2*R011 -  .5 *EA1*R021 +          R031
        R12 =          R012 +MULT2(EA3*R022)-      EA2*R032
        R22 =-.125*EA3*R012 +          R022 +  .25*EA1*R032
        R32 = .25 *EA2*R012 -      EA1*R022 +          R032
        R13 =          R013 +      EA3*R023 -MULT2(EA2*R033)
        R23 =-.25 *EA3*R013 +          R023 +      EA1*R033
        R33 = .125*EA2*R013 -  .25*EA1*R023 +          R033
C
C BRL POSITION ENTITIES.
C
        BP1 = .25 *X1 + EB1
        BP2 = .125*X2 + EB2
        BP3 = .125*X3 + EB3
C
        BD1 = .25 *XD1 + EDB1
        BD2 = .125*XD2 + EDB2
        BD3 = .125*XD3 + EDB3
C
        B1 =.1*BP1 + B001
        B2 =   BP2 + B002
        B3 =   BP3 + B003
C
C SOLVE FOR IX.
C
        DUM = .5 * (EA2*EA2 + EA3*EA3)
C
        IX1 = DUM*R0011
        IX2 = DUM*R0012
        IX3 = DUM*R0013
C
C SOLVE FOR IXD
C
```

```
         TEMP5 = -EA3*EDA3 - EA2*EDA2
C
C THE PHYSICAL EQUATIONS FOR THE NEXT TWO STATEMENTS ARE
C
C        TEMP6 = 2 * EA1 * EDA2
C        TEMP7 = 2 * EA1 * EDA3
C
         TEMP6 =  EA1*EDA2
         TEMP7 =  EA1*EDA3
C
         IXD1 =     R0011*TEMP5 + R0021*TEMP6 + R0031*TEMP7
         IXD2 = .25*R0012*TEMP5 + R0022*TEMP6 + R0032*TEMP7
         IXD3 = .25*R0013*TEMP5 + R0023*TEMP6 + R0033*TEMP7
C
C SOLVE FOR IXDD.
C
         DUM    =   MULT2 ( .625*(EA3*EDDA3 +EA2*EDDA2) )
         TEMP8  =  .5*EDA3*EDA3 + .5*EDA2*EDA2 + DUM
         TEMP9  = - EDA1*EDA2
         TEMP10 = - EDA1*EDA3
C
         IXDD1 = R0011*TEMP8 + .25*R0021*TEMP9 + .25*R0031*TEMP10
         IXDD2 = R0012*TEMP8 +     R0022*TEMP9 +     R0032*TEMP10
         IXDD3 = R0013*TEMP8 +     R0023*TEMP9 +     R0033*TEMP10
C
C 1.6 SHANK TO BLADE TRANSFORMATIONS
C
C AERO VELOCITY
C
C   CORRECTIONS TO THE AERO VELOCITY DUE TO INTERFERENCE EFFECTS.
C   NORMALLY THESE ARE PART OF THE "AIR MASS" SUBSYSTEM. FOR CONVENIENCE,
C   THESE HAVE INCLUDED IN THE ROTOR MODEL. NOTE THAT INSTEAD OF COMPUTING
C   AN AUGMENTIVE AERO VELOCITY, VAA(I), FOR CORRECTING VA(I), THE ACTUAL
C   AERO VELOCITY, VACT(I) IS COMPUTED IN THE FOLLOWING:
C
         VACT1=VAS1
         VACT2=VAS2
         VACT3=VAS3
C
C   INTRODUCE TOWER SHADOW AND WIND SHEAR EFFECTS ON AIR VELOCITY.
C   IF THE SHADOW REGION IS DEFINED AS A SECTOR (I.E. AN ANGLE),
C   THE SHADOW EFFECT CAN BE INTRODUCED OUTSIDE THE RADIAL LOOP.
C
C   IN GENERAL, USE FUNCTION SUBPROGRAMS "SHADOW" AND "SHEAR" TO GET
C   THE MODULATION COEFFICIENTS, SHDFAC AND SHRFAC BY TABLE LOOK-UP.
C
C        SHD = SHADOW ( S, SYI, CYI ) : ZERO => NO SHADOW EFFECT.
C        SHR = SHEAR  ( S, SYI, CYI ) : ZERO => NO SHEAR  EFFECT.
C
C   HERE, THE MODULATION COEFFICIENTS ARE COMPUTED BY EQUATIONS.
C
C   TOWER SHADOW => 28 PER CENT WITHIN A SECTOR OF 30 DEGREES :
C                  ( SHADOW = .28, AND SHDSEC = COS (15 DEG) ).
C
C   WIND  SHEAR  => 15 PER CENT : ( SHEAR = .15 ).
```

```
C
C
        SHR = SHEAR * S * CYI
        IF (SHR .EQ. 0) GO TO 1610
        VACT3 = VACT3 - SHR * VACT3
C
 1610 CONTINUE
C
C TO AVOID OVER/UNDER FLOWS AN ARBITRARY FACTOR IS USED IN "DUM= ... "
C
        DUM = .4*CYI - .4*SHDSEC
        IF (DUM .LT. 0   ) GO TO 1620
        SHD = SHADOW
        IF (SHD .EQ. 0) GO TO 1620
        VACT3 = (1. - SHD) * VACT3

C
 1620 CONTINUE
C
C  THE ACTUAL AERO VELOCITY SEEN BY THE BLADE IS VACT(I).
C
        TEMP11 =    VACT1 + .5*BD1 + .2*WAS2*B3 -     WAS3*B2
        TEMP12 = .2*VACT2 + .2*BD2 +    WAS3*B1 -. 04*WAS1*B3
        TEMP13 =    VACT3 +    BD3 + .2*WAS1*B2 -     WAS2*B1
C
        TEMP = MULT8(.625*R32*TEMP12)
C
        VA1 = .4*R11*TEMP11 + R12  *TEMP12     + .2*R13 *TEMP13
        VA2 = .1*R21*TEMP11 +MULT2(R22*TEMP12) + .2*R23 *TEMP13
        VA3 = .5*R31*TEMP11 + TEMP           +MULT2(R33 *TEMP13)
C
C
C 1.7 DISTRIBUTED INERTIAL LOADS
C
C THE PHYSICAL EQUATIONS FOR THE NEXT THREE STATEMENTS ARE,
C
C       TEMP14 = 2 * ( -WSMZ3*BD2 + WSMZ2*BD3 + Z*XD2 )
C       TEMP15 = 2 * (  WSMZ3*BD1 - WSMZ1*BD3 - Z*XD1 )
C       TEMP16 = 2 * (  WSMZ1*BD2 - WSMZ2*BD1 )
C
        TEMP14 =-.8*WSMZ3*BD2 + MULT2(.8*WSMZ2*BD3 + Z*XD2)
        TEMP15 = .4*WSMZ3*BD1 - MULT2(.8*WSMZ1*BD3 + Z*XD1)
        TEMP16 = .4* WSMZ1*BD2 - .2*WSMZ2*BD1
C
        UMBR1 = US1 + XDD1 + TEMP14 +.8*WDS2*B3 - .4*WDS3*B2
      @    + MULT4(U11*B1) +.8*(.1*U12*B2 + U13*B3) - ZSQ*X1
C
        UMBR2 = US2 + XDD2 + TEMP15 + MULT2(WDS3*B1) - .8*WDS1*B3
      @        +.4*U12*B1 + .8*(U22*B2 + U23*B3) - ZSQ*X2
C
        UMBR3 = .25*(US3+XDD3) + TEMP16 + .2*WDS1*B2 - WDS2*B1
      @          + U13*B1 + .2*(U23*B2 + .1*U33*B3)
C
        FMI1 = - MULT2 (MB*UMBR1)
        FMI2 = - MB*UMBR2
        FMI3 = - MULT2 (.8*MB*UMBR3)
C
C
C 1.8  AERO GEOMETRY
C
        M2 = MULT2(VA2*RVC)
        M3 = MULT2(VA3*RVC)
C
C
C 1.9  AERODYNAMIC COEFFICIENT TABLES
C
C 1.10 DISTRIBUTED AIR LOADS
C
C---- SIMPLIFIED AERO MODEL, WITH THE FOLLOWING ASSUMPTIONS:
C
C     LIFT COEFFICIENT ....... CFA3 = 6
C     DRAG COEFFICIENT ....... CFA2 = 0.01
C     INFLOW ANGLE = TEMP13/VA2 = MNOR/M2
C     ANGLE OF ATTACK ........ M3/M2
C     BLADE PRECONE ANGLE .... 0.12 RADIANS
C
C THE PHYSICAL EQUATIONS ARE,
C
C     LIFT  = 8886 * M2 * M3 * CB * PR
C     LIFTX = 8886 * M3 * MNOR * CB * PR; LIFT COMPONENT PRODUCING TORQUE.
C     DRAG  = 14.81 * M2 * M2 * CB * PR
C
      MNOR = MULT2(TEMP13*RVC)
C
      LIFT  = MULT4 ( .7075 * MULT2(M2*CB) * M3 * PR )
C
      LIFTX = MULT4 ( .7075 * MNOR * M3 * PR * CB )
C
C TIP LOSS CORRECTION: EQUIVALENT TO AN EFFECTIVE RADIUS OF 97 %.
C
      IF (TIPLOS .EQ. 0) GO TO 1110
      IF (J .NE. 11) GO TO 1110
      LIFT  = .4 * LIFT
      LIFTX = .4 * LIFTX
 1110 CONTINUE
C
      DRAG  = .168 * M2*M2*PR*CB
C
      FMA1 = .6 * LIFT - FA01
      FMA2 = - DRAG + LIFTX - FA02
      FMA3 = - LIFT - FA03
      FMA4 = .2 * QAC * CMA * CB
C
C 1.11 LOADS INTEGRAND ASSEMBLER.
C
        FMB1 = FMI1 + FMA1
        FMB2 = FMI2 + FMA2
        FMB3 = FMI3 + FMA3
C
        IGR1 = GP11*FMB1 + GP12*FMB2 + GP13*FMB3
        IGR2 = GP21*FMB1 + GP22*FMB2 + GP23*FMB3
        IGR3 = GP34*FMA4
C
        IFR1 = FP11*FMB1 + FP12*FMB2 + FP13*FMB3
        IFR2 = FP21*FMB1 + FP22*FMB2 + FP23*FMB3
        IFR3 = FP31*FMB1 + FP32*FMB2 + MULT2(FP33*FMB3)
C
C
C THE PHYSICAL EQUATIONS FOR THE NEXT TWO STATEMENTS ARE,
C
C       TEMP21 = EDDB1 + 2.*Z*EDB2 - ZSQ*EB1
C       TEMP22 = EDDB2 - 2.*Z*EDB1 - ZSQ*EB2
C
        TEMP21 = .5*EDDB1 + .8*Z*EDB2 - .2*ZSQ*EB1
        TEMP22 =    EDDB2 - .4*Z*EDB1 - .4*ZSQ*EB2
C
        DUM8 = -MULT4 (MB*TEMP21)
        DUM9 = -MULT4 (MB*TEMP22)
        DUM10= -MULT4 (MB*EDDB3)
C
C THE MB APPEARING IN THE NEXT TWO STATEMENTS IS ACTUALLY "MB*ZO*ZO"
C
        TEMP23 = MB*B001 + .05*FA01
        TEMP24 = MB*B002 + .5 *FA02
C
        IMR1 = MP11*FMB1 + MP12*FMB2 + MP13*FMB3
     @   + .4 *(.5*BP2*FMB3 - .2*BP3*FMB2  - BP3*DUM9 + BP2*DUM10)
     @   + .025*(X2*FA03 - .8*X3*TEMP24)
C
        IMR2 = MP21*FMB1 + MP22*FMB2 + MULT2(MP23*FMB3)
     @   + .5 *(.2*BP3*FMB1 - .5*BP1*FMB3  - BP1*DUM10) + BP3*DUM8
     @   + .25*(X3*TEMP23-.25*X1*FA03)
C
        IMR3 = MP31*FMB1 + MP32*FMB2 + MP33*FMB3
     @   + .5 *(.2*BP1*FMB2 - .2*BP2*FMB1 + BP1*DUM9) - BP2*DUM8
     @   + .25*(.2*X1*TEMP24 - X2*TEMP23)
C
C 1.12 RADIAL SUMMATIONS.
C
        X1 = X1 + DS * IX1
        X2 = X2 + DS * IX2
        X3 = X3 + DS * IX3
C
        XD1 = XD1 + DS * IXD1
        XD2 = XD2 + DS * IXD2
        XD3 = XD3 + DS * IXD3
C
        XDD1 = XDD1 + DS * IXDD1
        XDD2 = XDD2 + DS * IXDD2
        XDD3 = XDD3 + DS * IXDD3
C
        GRI1 = GRI1 + DS*IGR1
        GRI2 = GRI2 + DS*IGR2
        GRI3 = GRI3 + DS*IGR3
C
```

```
        FRI1 = FRI1 + DS*IFR1
        FRI2 = FRI2 + DS*IFR2
        FRI3 = FRI3 + DS*IFR3
C
        MRI1 = MRI1 + DS*IMR1
        MRI2 = MRI2 + DS*IMR2
        MRI3 = MRI3 + DS*IMR3
C
C------------ RADIAL SWEEP DELAY LOOP -------------------------
C
 9005 CONTINUE
      DELAY1 = DELAY1 - DELAY2
      IF (DELAY1 .GT. 0) GO TO 9005
      DELAY1 = DELAY3
C
C--------------------------------------------------------------
C
C       TEST FOR COMPLETION OF RADIAL STATIONS.
C
        IF (J .LT. NR) GO TO 9002
C
C*****************  END OF RADIAL LOOP  **********************
C
C CLEAN UP THE RADIAL SUMMATIONS.
C
        X2 = .5 * X2
        X3 = .5 * X3
C
        XD2 = MULT2(XD2)
        XD3 = MULT2(XD3)
C
        XDD1 = MULT2(XDD1)
C
        FRI3 = .1 * FRI3
C
        MRI1 = .25 * MRI1
        MRI2 = .1  * MRI2
        MRI3 = .4  * MRI3
C
C-----  SAVE FORCES AND MOMENTS OF BLADE # 1 FOR
C       COMPUTATION OF ROOT LOADS IN SHANK AXES.
C
        IF (I .GT. 1) GO TO 1130
        FTEM1 = FRI1
        FTEM2 = FRI2
        FTEM3 = FRI3
        MTEM1 = MRI1
        MTEM2 = MRI2
        MTEM3 = MRI3
C
C
C 1.13 MODAL COORDINATE INTEGRATIONS
C
 1130 CONTINUE
C
```

```
        DUM11=ZP*QDI1
        DUM12=ZP*QDI2
        DUM13=ZP*QDI3
C
        DUM21=ZP*QI1
        DUM22=ZP*QI2
        DUM23=ZP*QI3
C
        TEMP25=ZZ*QI1
        TEMP26=ZZ*QI2
        TEMP27=ZZ*QI3
C
C GENERALIZED MODAL EXCITATIONS
C
        GI1=GRI1+GV11*QDI1+GZQ11*TEMP25+GZV11*DUM11
     @          +GV12*QDI2+GZQ12*TEMP26+GZV12*DUM12
     @          +GV13*QDI3+GZQ13*TEMP27+GZV13*DUM13+ZZ*GB01
C
        GI2=GRI2+GV21*QDI1+GZQ21*TEMP25+GZV21*DUM11
     @          +GV22*QDI2+GZQ22*TEMP26+GZV22*DUM12
     @          +GV23*QDI3+GZQ23*TEMP27+GZV23*DUM13+ZZ*GB02
C
        GI3=GRI3+GV31*QDI1+GZQ31*TEMP25+GZV31*DUM11
     @          +GV32*QDI2+GZQ32*TEMP26+GZV32*DUM12
     @          +GV33*QDI3+GZQ33*TEMP27+GZV33*DUM13+ZZ*GB03
C
        DUM24=ZP*GI1
        DUM25=ZP*GI2
        DUM26=ZP*GI3
C
C MODE # 1 ENTITIES
C
        QSI1   = QI1
        QI1    = KQQ11*QSI1   + KQV11*QDI1   + KQG11*GI1
     @         + KQZQ11*DUM21 + KQZV11*DUM11 + KQZG11*DUM24
        QDI1   = KVQ11*QSI1   + KVV11*QDI1   + KVG11*GI1
     @         + KVZQ11*DUM21 + KVZV11*DUM11 + KVZG11*DUM24
        QDDI1  = .8*GI1 - ESQQ11*QI1
C
C MODE # 2 ENTITIES
C
        QSI2   = QI2
        QI2    = KQQ22*QSI2   + KQV22*QDI2   + KQG22*GI2
     @         + KQZQ22*DUM22 + KQZV22*DUM12 + KQZG22*DUM25
        QDI2   = KVQ22*QSI2   + KVV22*QDI2   + KVG22*GI2
     @         + KVZQ22*DUM22 + KVZV22*DUM12 + KVZG22*DUM25
        QDDI2  = MULT2 (.4*GI2 - ESQQ22*QI2)
C
C MODE # 3 ENTITIES
C
        QSI3   = QI3
        QI3    = KQQ33*QSI3   + KQV33*QDI3   + KQG33*GI3
     @         + KQZQ33*DUM23 + KQZV33*DUM13 + KQZG33*DUM26
        QDI3   = KVQ33*QSI3   + KVV33*QDI3   + KVG33*GI3
     @         + KVZQ33*DUM23 + KVZV33*DUM13 + KVZG33*DUM26
```

```
       QDDI3 = 0
C
C SAVE MODAL STATE VARIABLES
C
C      CALL STORQS(I)
C
C THE FOLLOWING SEQUENCE IMPLEMENTS THE "STORQS" FUNCTION FOR 2 BLADES
C
C ALSO SAVE THE GENERALISED FORCING FUNCTION FOR BLADE # 1
C
  501  IF (I .GT. 1) GO TO 502
       QI1A=QI1
       QDI1A=QDI1
       QDDI1A=QDDI1
       QI2A=QI2
       QDI2A=QDI2
       QDDI2A=QDDI2
       QI3A=QI3
       QDI3A=QDI3
       QDDI3A=QDDI3
       GO TO 503
C
  502 CONTINUE
       QI1B=QI1
       QDI1B=QDI1
       QDDI1B=QDDI1
       QI2B=QI2
       QDI2B=QDI2
       QDDI2B=QDDI2
       QI3B=QI3
       QDI3B=QDI3
       QDDI3B=QDDI3
C
  503  CONTINUE
C
C 1.14 MULTIBLADE MODAL COMBINATIONS
C
       SYBION=RNB*SYBI
       CYBION=RNB*CYBI
C
       QS1=QS1+QI1*SYBION
       QC1=QC1+QI1*CYBION
       Q01=Q01+QI1*RNB
C
       QDS1=QDS1+QDI1*SYBION
       QDC1=QDC1+QDI1*CYBION
       QD01=QD01+QDI1*RNB
C
       QS2=QS2+QI2*SYBION
       QC2=QC2+QI2*CYBION
       Q02=Q02+QI2*RNB
C
       QDS2=QDS2+QDI2*SYBION
       QDC2=QDC2+QDI2*CYBION
       QD02=QD02+QDI2*RNB
```

```
C
        QS3=QS3+QI3*SYBION
        QC3=QC3+QI3*CYBION
        Q03=Q03+QI3*RNB
C
        QDS3=QDS3+QDI3*SYBION
        QDC3=QDC3+QDI3*CYBION
        QD03=QD03+QDI3*RNB
C
C 1.15 MULTIBLADE LOADS COMBINATIONS
C
        FGR1 = FGR1 + CYBION*FRI1 + SYBION*FRI2
        FGR2 = FGR2 - SYBION*FRI1 + CYBION*FRI2
        FGR3 = FGR3 + RNB*FRI3
C
        MGR1 = MGR1 + CYBION*MRI1 + SYBION*MRI2
        MGR2 = MGR2 - SYBION*MRI1 + CYBION*MRI2
        MGR3 = MGR3 + RNB*MRI3
C
C
C     TEST FOR COMPLETION OF BLADES.
C
*     IF (I .LT. NB) GO TO 9001
C
C*****************  END OF BLADE LOOP  ***********************
C
C
C 1.16 GIMBAL LOADS SUMMATION
C
        FSQ11=FQ11+FZQ11*ZZ
        FSQ21=FQ21+FZQ21*ZZ
        FSQ31=FQ31+FZQ31*ZZ
C
        FSQD11=FQD11+FZQD11*ZP
        FSQD21=FQD21+FZQD21*ZP
        FSQD31=FQD31+FZQD31*ZP
C
        MSQ11=MQ11+MZQ11*ZZ
        MSQ21=MQ21+MZQ21*ZZ
        MSQ31=MQ31+MZQ31*ZZ
C
        MSQD11=MQD11+MZQD11*ZP
        MSQD21=MQD21+MZQD21*ZP
        MSQD31=MQD31+MZQD31*ZP
C
        FSQ12=FQ12+FZQ12*ZZ
        FSQ22=FQ22+FZQ22*ZZ
        FSQ32=FQ32+FZQ32*ZZ
C
        FSQD12=FQD12+FZQD12*ZP
        FSQD22=FQD22+FZQD22*ZP
        FSQD32=FQD32+FZQD32*ZP
C
        MSQ12=MQ12+MZQ12*ZZ
        MSQ22=MQ22+MZQ22*ZZ
        MSQ32=MQ32+MZQ32*ZZ
C
        MSQD12=MQD12+MZQD12*ZP
        MSQD22=MQD22+MZQD22*ZP
        MSQD32=MQD32+MZQD32*ZP
C
        FSQ13=FQ13+FZQ13*ZZ
        FSQ23=FQ23+FZQ23*ZZ
        FSQ33=FQ33+FZQ33*ZZ
C
        FSQD13=FQD13+FZQD13*ZP
        FSQD23=FQD23+FZQD23*ZP
        FSQD33=FQD33+FZQD33*ZP
C
        MSQ13=MQ13+MZQ13*ZZ
        MSQ23=MQ23+MZQ23*ZZ
        MSQ33=MQ33+MZQ33*ZZ
C
        MSQD13=MQD13+MZQD13*ZP
        MSQD23=MQD23+MZQD23*ZP
        MSQD33=MQD33+MZQD33*ZP
C
        FS01 = F01 + MULT2(FZ1*ZZ)
        FS02 = F02 + FZ2*ZZ
        FS03 = F03 + FZ3*ZZ
        MS01 = M01 + MZ1*ZZ
        MS02 = M02 + MZ2*ZZ
        MS03 = M03 + MZ3*ZZ
C
        FG1=FGR1+FSQ11*QC1+FSQ21*QS1 +FSQD11*QDC1+FSQD21*QDS1
     @         + FSQ12*QC2+MULT4(FSQ22*QS2)+FSQD12*QDC2+FSQD22*QDS2
     @          +FSQ13*QC3+FSQ23*QS3 +FSQD13*QDC3+FSQD23*QDS3
C
        FG2=FGR2+FSQ21*QC1 -FSQ11*QS1+FSQD21*QDC1-FSQD11*QDS1
     @          +MULT4(FSQ22*QC2)-FSQ12*QS2+FSQD22*QDC2-FSQD12*QDS2
     @          +FSQ23*QC3 -FSQ13*QS3+FSQD23*QDC3+FSQD13*QDS3
C
        FG3=FGR3+FSQ31*Q01+FSQD31*QD01
     @          +FSQ32*Q02+FSQD32*QD02
     @          +FSQ33*Q03+FSQD33*QD03 + FS03
C
        MG1=MGR1+MSQ11*QC1+MSQ21*QS1+MSQD11*QDC1+MSQD21*QDS1
     @          +MSQ12*QC2+MSQ22*QS2+MSQD12*QDC2+MSQD22*QDS2
     @          +MSQ13*QC3+MSQ23*QS3+MSQD13*QDC3+MSQD23*QDS3
C
        MG2=MGR2+MSQ21*QC1-MSQ11*QS1+MSQD21*QDC1-MSQD11*QDS1
     @          +MSQ22*QC2-MSQ12*QS2+MSQD22*QDC2-MSQD12*QDS2
     @          +MSQ23*QC3-MSQ13*QS3+MSQD23*QDC3-MSQD13*QDS3
C
        MG3=MGR3+MSQ31*Q01+MSQD31*QD01
     @          +MULT2(MSQ32*Q02)+MSQD32*QD02
     @          +MSQ33*Q03+MSQD33*QD03
C
C-------------------------------------------------------------------
C COMPUTE CORRECTIONS TO THE FLATWISE AND EDGEWISE MODAL QUANTITIES
```

```
C THESE CORRECTIONS WERE ADDED TO THE MOD-0 WIND TURBINE MODEL TO
C ELIMINATE NUMERICAL INSTABILITIES WHICH AROSE WHEN THE ROTOR MODEL
C WAS COUPLED TO THE POWER TRAIN AND THE TOWER MODELS.
C---------------------------------------------------------------------
C
      IF (TOWERX .EQ. 0) GO TO 6010
C
C FLATWISE MODE:
C
C CORRECTION TO THE ACCELERATION
C
      QDD1X = .1 * (- .596 * FG3 + .621 * SDD1 + .332 * S1)
C
C CLEAN UP THE FLATWISE MODAL QUANTITIES
C
      QDDI1A = QDDI1A + QDD1X
      QDDI1B = QDDI1B + QDD1X
C
      QDSAVA = QDI1A
      QDSAVB = QDI1B
      QDI1A  = QDI1A + DTQ * QDD1X
      QDI1B  = QDI1B + DTQ * QDD1X
C
      QI1A = QI1A + .4 * DTQ * (QDI1A-QDSAVA)
      QI1B = QI1B + .4 * DTQ * (QDI1B-QDSAVB)
C
C COMPUTE THE CORRECT THRUST
C
      FG3 = FG3 + .7453 * QDD1X
C
 6010 CONTINUE
C
C
      IF (PWRTRN .EQ. 0) GO TO 6020
C
C EDGEWISE MODE:
C
C CORRECTION TO THE ACCELERATION
C
      MG3X = - MULT2 (MG3)
      QDD2X = - MULT4 (.819*MG3X - .655*MPOW) + .0453 * ZD
C
C CLEAN UP THE EDGEWISE MODAL QUANTITIES
C
      QDDI2A = QDDI2A + QDD2X
      QDDI2B = QDDI2B + QDD2X
C
      QDSAVA = QDI2A
      QDSAVB = QDI2B
      QDI2A  = QDI2A + DTQ * QDD2X
      QDI2B  = QDI2B + DTQ * QDD2X
C
      QI2A = QI2A + .4 * DTQ * (QDI2A-QDSAVA)
      QI2B = QI2B + .4 * DTQ * (QDI2B-QDSAVB)
C
```

```
C COMPUTE THE CORRECT TORQUE
C
        MG3 = - ( .5*MG3X  + .1446*QDD2X)
C
 6020   CONTINUE
C
C
C-----------------------------------------------
C COMPUTE ROOT LOADS OF BLADE # 1 IN SHANK AXES
C-----------------------------------------------
C
        FBLD1 = FTEM1 + FSQ11*QI1A + FSQD11*QDI1A
     @                + FSQ12*QI2A + FSQD12*QDI2A
     @                + FSQ13*QI3A + FSQD13*QDI3A
C
        FBLD2 = FTEM2 + FSQ21*QI1A + FSQD21*QDI1A
     @           +MULT4(FSQ22*QI2A)+ FSQD22*QDI2A
     @                + FSQ23*QI3A + FSQD23*QDI3A + FS02
C
        FBLD3 = FTEM3 + FSQ31*QI1A + FSQD31*QDI1A
     @                + FSQ32*QI2A + FSQD32*QDI2A
     @                + FSQ33*QI3A + FSQD33*QDI3A + FS03
C
        MBLD1 = MTEM1 + MSQ11*QI1A + MSQD11*QDI1A
     @                + MSQ12*QI2A + MSQD12*QDI2A
     @                + MSQ13*QI3A + MSQD13*QDI3A + MS01
C
        MBLD2 = MTEM2 + MSQ21*QI1A + MSQD21*QDI1A
     @                + MSQ22*QI2A + MSQD22*QDI2A
     @                + MSQ23*QI3A + MSQD23*QDI3A + MS02
C
        MBLD3 = MTEM3 + MSQ31*QI1A + MSQD31*QDI1A
     @           +MULT2(MSQ32*QI2A)+ MSQD32*QDI2A
     @                + MSQ33*QI3A + MSQD33*QDI3A + MS03
C
C
C 1.17 GIMBAL TO HUB TRANSFORMATIONS
C
        FCU1 = CYC*FG1 + SYC*FG2
        FCU2 =-SYC*FG1 + CYC*FG2
C
        MCU1 = CYC*MG1+SYC*MG2 - .025*ZU*FCU2
        MCU2 =-SYC*MG1+CYC*MG2 + .025*ZU*FCU1
C
        EG1 = MCU2
C
        FT1 = FCU1 + MULT2(QG1*FG3)
C
        MT1 = MCU1 + .05*QG1*MG3 + .025*ZU*FCU2
        MT2 = MCU2 - .025*ZU*FT1
C
        FH1 = CYT*FT1 - SYT*FCU2
        FH2 = SYT*FT1 + CYT*FCU2
        FH3 = MULT2 (FG3 - .02*QG1*FCU1)
C
```

```
      MH1 = CYT*MT1 - SYT*MT2
      MH2 = SYT*MT1 + CYT*MT2
      MH3 = MG3 - .8*QG1*MCU1
C
C 1.18 GIMBAL SERVO INTEGRATOR.
C
      QDDG1=KGE*EG1+KGQC*QC1+KGQS*QS1
     @          +KGVC*QDC1+KGVS*QDS1
C
      QSG1=QG1
C
      DUM14=ZP*QDG1
      DUM15=ZP*QDDG1
      DUM16=ZP*QSG1
      DUM17=ZP*QDDG1
C
      QG1=GQQ*QSG1+GQV*QDG1+GQZV*DUM14
     @          +GQA*QDDG1+GQZA*DUM15
C
      QDG1=GVV*QDG1+GVQ*QSG1+GVZQ*DUM16
     @          +GVA*QDDG1+GVZA*DUM17
C
C
C     END THE ROTOR MODEL.
C
```

```
C
C 2.0 SUPPORT SYSTEM EQUATIONS.
C
        IF (TOWER .EQ. 0) GO TO 2010
C
C MODEL WITH THREE NORMAL MODES FOR SURGE, SWAY, AND YAW(TORSION).
C
C GENERALIZED FORCING FUNCTIONS
C
        GS1 = GSF11*FH1 + GSF13*FH3 + GSM12*MH2
        GS2 = GSF22*FH2 + GSM23*MPOD
        GS3 = GSF32*FH2 + MULT2(GSM31*MH1)
C
C MODAL ACCELERATIONS
C
        SDD1 = GS1 - DS11*SD1 - W2S11*S1
        SDD2 = GS2 - DS22*SD2 - W2S22*S2
        SDD3 = GS3 - DS33*SD3 - W2S33*S3
C
C NUMERICAL INTEGRATION OF MODAL COORDINATES
C
        SD1 = SD1 + .2*SDD1*DTS
        SD2 = SD2 + .2*SDD2*DTS
        SD3 = SD3 + .4*SDD3*DTS
C
        S1 = S1 + SD1*DTS
        S2 = S2 + SD2*DTS
        S3 = S3 + SD3*DTS
C
C TRANSLATIONAL VELOCITY
C
        VH1 = VBH1 + SV11*SD1
        VH2 = VBH2 + SV22*SD2 + .5*SV23*SD3
        VH3 = VBH3 + SV31*SD1
C
C ROTATIONAL VELOCITY
C
        WH1 = WBH1 + .5*SW13*SD3
        WH2 = WBH2 +    SW21*SD1
        WH3 = WBH3 +    SW32*SD2
C
C TIME DERIVATIVE OF TRANSLATIONAL VELOCITY
C
        VDH1 = VDBH1 + SV11*SDD1
        VDH2 = VDBH2 + SV22*SDD2 + SV23*SDD3
        VDH3 = VDBH3 + SV31*SDD1
C
C TIME DERIVATIVE OF ROTATIONAL VELOCITY
C
        WDH1 = WDBH1 + SW13*SDD3
        WDH2 = WDBH2 + SW21*SDD1
        WDH3 = WDBH3 + SW32*SDD2
C
 2010   CONTINUE
C
```

```
C
C 3.0-CONTROL SYSTEM
C
      IF (CONTRL .EQ. 0) GO TO 3010
C
      ZERR = MULT4 (SRQ - Z)
      AD1  = AZ1 * ZERR
      AD2  = AA21*A1 - AA22*A2 + AZ2 * ZERR
      AD3  = AG3*ZA + AD03
C
C NUMERICAL INTEGRATION
C
      A1  = A1 + DTC1 * AD1
      A2  = A2 + MULT2 (DTC2 * AD2)
      A3  = A3 + DTC3 * AD3
C
C RESULTS
C
      C1  = CA12*A2
      C2  = 0
      C3  = 0
C
 3010 CONTINUE
C
      LGC = MULT2 (LA13*A3 + LPR*PRQ + LG*ZA)
C
C
C 4.0 - POWER TRAIN
C
      IF (PWRTRN .EQ. 0) GO TO 4010
C
C THE EQUATIONS SHOWN HERE FOR THE POWER TRAIN MODEL ARE THE SPECIALIZED
C FORMS USED IN THE MOD-0 SIMULATION. THESE SPECIAL FORMS WERE NEEDED TO
C ELIMINATE THE NUMERICAL INSTABILTY OBSERVED DURING THE COUPLING OF THE
C ROTOR AND THE POWER TRAIN MODELS.
C
C ACCELERATIONS.
C
      PDD1 = ZD - MULT2(.5787*QDD2X)
      PDD2 = .1*PDD1-.5674*MPOW+BP21*PD1-BP22*PD2+ MULT2(.7095*LGC)
C
C NUMERICAL INTEGRATION.
C
      PD1 = PD1 + PDD1 * DTP1
      PD2 = PD2 + PDD2 * DTP2
      P2  = P2  + .5*PD2  * DTP2
C
C FIND THE TORQUES APPLIED BY THE POWER TRAIN TO THE ROTOR & THE TOWER
C
      MPOW = MULT2 (.784 * P2) + BPOW1 * PD1 + BPOW2 * PD2
      MPOD = - BPOD1*PD1 + BPOD2*PD2 - LGC
C
C OUTPUTS
C
      ZD = PDD1
```

```
        Z  = PD1
        ZP= MULT2 (Z-.5)
        ZA = PD1 - .1*PD2
C
 4010   CONTINUE
C
C RATE RESOLVER:
C TO AVOID OVER/UNDER FLOW PROBLEMS COMPUTE ONE HALF OF THE SINE AND
C COSINE OF THE ROTOR AZIMUTHAL ANGLE, AND THEN DOUBLE THE RESULTS.
C
      SYHAF = .5*SY
      CYHAF = .5*CY
C
      SDY = ZP*DT + DT
C
C COMPUTE AMPLITUDE ERROR
C
      EA = SYHAF*SYHAF + CYHAF*CYHAF - .25
C
C MAKE CORRECTION
C
      CDY = CDY - .5*EA
C
C SET UP ADVANCE
C
      CYSAV = CYHAF
      CYHAF = (CYHAF*CDY+CYHAF*CDY) - SYHAF*SDY
      SYHAF = (SYHAF*CDY+SYHAF*CDY) + CYSAV*SDY
C
      CY = MULT2 (CYHAF)
      SY = MULT2 (SYHAF)
C
C
C 5.0 AIR MASS.
C
C RETARDATION VELOCITY
C
        IF (RETARD .EQ. 0) GO TO 5010
C
C   ERROR FUNCTION
C
        EVR = .04 - MULT4 ( RV*RV * (VAH2*VAH2 + VAH3*VAH3) )
C
C   CORRECTION SERVO
C
        RV = RV + MULT16 (EVR*GCEV)
C
C   RETARDATION
C
        VRW = KAR * FH3 * RV * RHOR
C
 5010   CONTINUE
C
C
C WINDGUST RANDOM SIGNALS
```

```
C
        IF (GUSTS .EQ. 0) GO TO 5020
C
C   EQUATIONS WHICH FOLLOW USE A RANDOM NUMBER GENERATOR WHICH
C   APPEARS AS A WHITE NOISE SOURCE TO EACH FILTER EQUATION.
C
        QDDGU1=RANDOM(IGNORE) -ZTG11*WGG11*QDGU1 -WGG11*WGG11*QGU1
        QDDGU2=RANDOM(IGNORE) -ZTG22*WGG22*QDGU2 -WGG22*WGG22*QGU2
        QDDGU3=RANDOM(IGNORE) -ZTG33*WGG33*QDGU3 -WGG33*WGG33*QGU3
        QDDGU4=RANDOM(IGNORE) -ZTG44*WGG44*QDGU4 -WGG44*WGG44*QGU4
        QDDGU5=RANDOM(IGNORE) -ZTG55*WGG55*QDGU5 -WGG55*WGG55*QGU5
        QDDGU6=RANDOM(IGNORE) -ZTG66*WGG66*QDGU6 -WGG66*WGG66*QGU6
C
C   NUMERICAL INTEGRATION OF WINDGUST FILTERS.
C
        QDGU1=QDGU1+QDDGU1*DTG
        QDGU2=QDGU2+QDDGU2*DTG
        QDGU3=QDGU3+QDDGU3*DTG
        QDGU4=QDGU4+QDDGU4*DTG
        QDGU5=QDGU5+QDDGU5*DTG
        QDGU6=QDGU6+QDDGU6*DTG
C
C
        QGU1=QGU1+QDGU1*DTG
        QGU2=QGU2+QDGU2*DTG
        QGU3=QGU3+QDGU3*DTG
        QGU4=QGU4+QDGU4*DTG
        QGU5=QGU5+QDGU5*DTG
        QGU6=QGU6+QDGU6*DTG
C
C   OUTPUTS
C
        VG1=VGU11*QDGU1
        VG2=VGU22*QDGU2
        VG3=VGU33*QDGU3
C
        WG1=WGU11*QDGU4
        WG2=WGU22*QDGU5
        WG3=WGU33*QDGU6
C
 5020   CONTINUE
C
C PRIMARY WIND SIGNALS
C
        VWT1=VW1+VG1
        VWT2=VW2+VG2
        VWT3=VW3+VG3
C
        WWT1=WW1+WG1
        WWT2=WW2+WG2
        WWT3=WW3+WG3
C
        VWT1=VWT1+VRW
C
C
```

```
C INTERFERENCE EFFECTS: FOR CONVENIENCE,  TOWER SHADOW AND WINDSHEAR
C HAVE BEEN INCLUDEDIN THE ROTOR MODEL.
C
C PASS THROUGH MODELS IS COMPLETE
C
C************  BOTTOM OF INFINITE LOOP  *******************
C
      GO TO 9000
```

1. Report No.
NASA CR-174983

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
WEST-3 Wind Turbine Simulator Development
Volume 3: Software

5. Report Date
July 1985

6. Performing Organization Code

7. Author(s)
J.A. Hoffman and S. Sridhar

8. Performing Organization Report No.
PPI-FID-300086

10. Work Unit No.

9. Performing Organization Name and Address
Paragon Pacific Inc.
530 Maple Ave.
Torrance, California 90503

11. Contract or Grant No.
DEN 3-247

13. Type of Report and Period Covered
Contractor Report

12. Sponsoring Agency Name and Address
U.S. Department of Energy
Conservation and Renewable Energy
Washington, D.C. 20545

14. Sponsoring Agency ~~Code~~Report No.
DOE/NASA/0247-3

15. Supplementary Notes
Final Report. Prepared under Interagency Agreement DE-AI01-76ET20320. Project Manger, D. Janetzke, Structures Division, NASA Lewis Research Center, Cleveland, Ohio 44135.

16. Abstract

This report deals with the software developed for WEST-3, a new, all digital, and fully programmable wind turbine simulator developed by Paragon Pacific Inc. The process of wind turbine simulation on WEST-3 is described in detail. The major steps are, the processing of the mathematical models, the preparation of the constant data, and the use of system software to generate executable code for running on WEST-3. The mechanics of reformulation, normalization, and scaling of the mathematical models is discussed in detail, in particular, the significance of reformulation which leads to accurate simulations.

Descriptions for the preprocessor computer programs which are used to prepare the constant data needed in the simulation are given. These programs, in addition to scaling and normalizing all the constants, relieve the user from having to generate a large number of constants used in the simulation. Also given in the report are brief descriptions of the components of the WEST-3 system software: Translator, Assembler, Linker, and Loader.

Also included in the report are: details of the aeroelastic rotor analysis, which is the center piece of a wind turbine simulation model; analysis of the gimbal subsystem; and listings of the variables, constants, and equations used in the simulation.

17. Key Words (Suggested by Author(s))
Wind turbine simulation
Parallel processing

18. Distribution Statement
Unclassified - unlimited
STAR Category 44
DOE Category UC-60

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price*
Unclassified	Unclassified	230	A11

*For sale by the National Technical Information Service, Springfield, Virginia 22161

☆U.S. GOVERNMENT PRINTING OFFICE: 1986-748-120-40408